ISMM

THE UNIVERSITY OF WEST FLORIDA
Pensacola, Florida

COMPUTER AIDED MODELING OF COGNITIVE PERFORMANCE ASSESSMENT TESTS USING THE MICROSAINT SOFTWARE

BY

S. SHAMMA

R. STANNY

R. LAUGHERY

C. PLATT

D. SHERRY

INSTITUTE FOR STATISTICAL AND MATHEMATICAL MODELING

FEBRUARY, 1989

March 1989

Computer Aided Modeling of Cognitive Performance Assessment Tests Using The MicroSAINT Software

C MIPR88PP8805
WU 63764A
3M4637648995.
AB-088

S. Shamma[1], R. Stanny, R. Laughery[2], C. Platt[2], and D. Sherry[1]

Naval Aerospace Medical Research Laboratory
Naval Air Station
Pensacola, Florida 32508-5700

Walter Reed Army Institute of Research
Washington, DC 20307-5100/
Naval Medical Research and Development Command
National Naval Medical Center, Bldg. 1
Bethesda, MD 20814-5044

(1) University of West Florida, Pensacola, FL
(2) Micro Analysis and Design, Inc., Boulder, CO

Approved for public release; distribution unlimited.

This report contains software documentation for a set of computer simulations developed, using the MicroSAINT software, to characterize human behavior in a subset of performance-assessment tests drawn from the Unified Tri-Services Cognitive Performance Assessment Battery (UTC-PAB). The goal of the effort is to characterize the cognitive functions that contribute to performance in the tests of the UTC-PAB. The purposes of this are to: (1) improve the prediction of stressor effects on human performance and (2) make it possible to generalize in principled fashion from performance on the tests of the UTC-PAB to performance in operational settings.

Human Performance, Human Factors, Modeling, Cognition

108

Unclassified Unclassified Unclassified

COMPUTER AIDED MODELING OF COGNITIVE PERFORMANCE ASSESSMENT TESTS USING THE MICROSAINT SOFTWARE

Shawky Shamma

Robert Stanny

Ronald Laughery

Chris Platt

David Sherry

Approved: Charles M. Bundrick

Charles M. Bundrick, Director

Date: 3/13/89

The following research staff, their associates and/or students have contributed to this report:

R. Laughery, Micro-SAINT Analysis and Design, Boulder, CO

C. Platt, Micro-SAINT Analysis and Design, Boulder, CO

S. Shamma, University of West Florida, Pensacola, FL

D. Sherry, University of West Florida, Pensacola, FL

R. Stanny, Naval Aerospace Medical Laboratory, Pensacola, FL

SUMMARY

This report documents a group of computer simulations that have been developed, using the MicroSAINT software, to characterize human behavior in a subset of performance assessment tests drawn from the Unified Tri-Services Cognitive Performance Assessment Battery (UTC PAB). The goal of the effort of which this is one part is to characterize the cognitive functions that contribute to performance on the tests of the UTC-PAB, to make it possible to predict the effects of stressors on performance in those tests, and to make it possible to generalize in principled fashion from performance on the tests of the UTC-PAB to performance in operational settings.

The programs documented here embody three, preliminary models developed in order to characterize performance in tests of spatial perception, choice reaction time, and logical (grammatical) reasoning, respectively. A brief orientation to each model is presented here along with the corresponding program documentation. A treatment of the theory and performance of each model will be published elsewhere.

Acknowledgements

This work was sponsored by the Joint Working Group on Drug Dependent Degradation in Military Performance under USAMRDC work unit No. 63764A 3M4637648995.AB-088 and conducted at the Naval Aerospace Medical Research Laboratory, Naval Air Station, Pensacola, FL. We would like to acknowledge Commander R.P. Olafson, MC, USNR, and Captain O.G. Blackwell, MC, USN, for their support. We would also like to express our appreciation to Mr. E.A. Molina, Lieutenant Commander R.P. Crisman, MSC, USN, and Lieutenant D.L. Reeves, MSC, USN for providing a stimulating working atmosphere during our visits to NAMRL. To these people, we are grateful for their encouragement and invaluable assistance. We also thank Mrs. Jean Payne for her special efforts in the preparation of this manuscript.

TABLE OF CONTENTS

FIGURES

I. Introduction

The Unified Tri-service Cognitive Performance Assessment Battery (UTC-PAB), [3,6] was a product of the Tri-Service Joint Working Group on Drug Dependent Degradation of Military Performance (JWGD3 MILPERF). The UTC-PAB represents the primary instrument for the assessment of cognitive performance in JWGD3 sponsored drug evaluation programs.

The premise for UTC-PAB development was to provide a standardized metric that is responsive to required military-mission abilities and skills and that will be a sensitive instrument for detecting performance decrements due to the use of biomedical treatment drugs.

The purpose of this report is to illustrate computer simulation of the performance of the subjects in some tests of UTC-PAB. MicroSAINT software [5] is used in the simulation and in analyzing the results. The goal of such study is to provide a structural framework for categorizing tests in UTC-PAB and simulating subjects' performance strategies in these tests. Such a framework may provide a mechanism for developing unified predictive models with respect to the effect of individual drugs on performance.

The following sections constitute a detailed summary and results of modeling three individual UTC-PAB tests:

a) Manikin test
b) Four choice
c) Logical reasoning

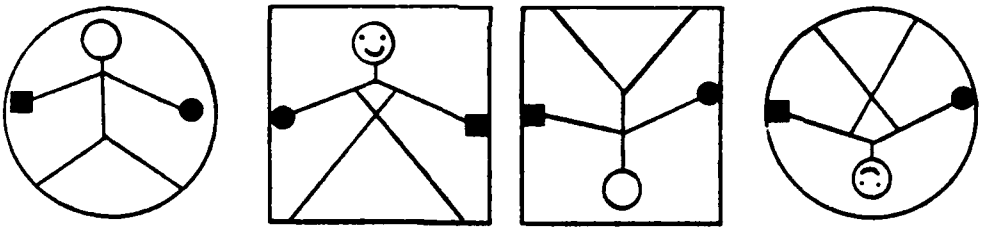
Each section starts with a brief description of the model-detailed information about the tests are in [3,6] - followed by a mix of model printouts from MicroSAINT and word processed explanations. At the end of each section are the Micro Saint printouts of the simulation scenario, snapshots, functions, continuous variable changes as appropriate, and summary data outputs and results.

II. Manikin Test Models

In performing the manikin test, the subject may adopt a variety of distinct strategies or may mix several of these strategies. Three models are constructed corresponding to three different strategies that may be considered by the subjects; we have assumed that only one strategy is used in a consistent fashion in each model. These three models may be used to reflect other strategies by changing variable values in some cases or by changing the nature of the MicroSAINT task network in others. The results of these models and other related models will need to be compared to the empirical data to determine which ones are most appropriate. The three chosen models are based on a determination by the subject of the manikin orientation relative to his desired (preferred) internal orientation(s).

The following figure and tables show sketches for the four basic manikin orientations and the three models corresponding to a maximum of two, one, or no rotations of the manikin.

Figure 1. Different Manikin Orientations



	Back Facing Subject	Face Facing Subject	Upside Down Back Facing Subject	Upside Down Face Facing Subject
Manikin Number	1	2	3	4

Manikin 1 Model

"Preferred" position is Manikin number 1 because the manikin's right hand is on the same side of the subject's right hand. We assume the subject rotates an internal representation of the Manikin to this position before solving the Manikin problem. A maximum of two rotations may be required to transform the manikin to the preferred position. The following table reflects the number of rotations required for transforming a given manikin to the preferred position.

Table 1

Manikin No.	Up-Down Match	Front-Back Match	# of Rotations	Axis of Rotation
1	yes	yes	0	none
2	yes	no	1	vertical axis
3	no	yes	1	front-back axis
4	no	no	2	front-back axis then vertical axis

Manikin 2 Model

Internal preferred position is Manikin number 1. A maximum of one rotation may be required to transform the manikin to the preferred position.

Table 2

Manikin No.	Up-Down Match	Front-Back Match	# of Rotations	Axis of Rotation
1	yes	yes	0	none
2	yes	no	1	vertical axis
3	no	yes	1	front-back axis
4	no	no	1	side axis (flip)

Manikin 3 Model

Internal preferred positions are Manikin number 1 and number 4. The subject uses a pattern match to these two positions (no rotations) because the manikin's right hand is on the same side of the subject's right hand.

Table 3

Manikin No.	Up-Down Match	Front-Back Match	Match (Preferred Position)
1	yes	yes	yes
2	yes	yes	no
3	no	yes	no
4	no	no	yes

The process of modeling is described by the following sequence of tasks. The stimulus is presented and processed through the sensory system to the cognitive centers of the brain. The subject then determines the orientation of the manikin presented in the stimulus relative to his desired internal orientation(s). This puts the manikin in the same position as the subject so that the determination of which hand the target object is in is straight forward. If the manikin is already in the desired orientation, the model proceeds to the next decision making step. If the manikin is not in the desired orientation then it is rotated according to table 1 or 2 or matched according to table 3 respectively. Once the orientation is correct, the subject then matches the target to the surrounding. That is, if the manikin is surrounded by a circle, the circle is identified as the target. The hand which contains the target object is then identified and the motor response is made. This completes modeling one trial of the test.

Two types of errors are modeled. The first one occurs if the subject forgets the orientation of the manikin or realizes that he has made a mistake in orienting the manikin. In this case he goes back to the beginning and starts over. The result is a time penalty and no accuracy penalty. The second type of error is an accuracy error. The subject makes a mistake and doesn't realize it. It is assumed that these types of errors can occur in the orientation matching step, the target-surround matching step, the hand identification step, and the motor response step. These errors are assumed to be independent so that they may correct one another.

Model Printouts and Word-Processed Explanations

(i) Manikin 1 Model

TASK NETWORK

Network Number: 0
(1) Name: manikin1 (2) Type: Network
(3) Upper Network:
(4) Release Condition: 1;
(5) First sub-job: start start
(6) Sub-jobs (each can be task or network):

Number:	Name:	Type:
stim	stimulus	Task
sens	sensory processing	Task
omatch	match orientation	Task
rotate	rotate manakin	Task
tsmatch	targ. surround match	Task
idhand	ID hand with object	Task
motor	motor response	Task
right	right response	Task
wrong	wrong response	Task
start	start	Task

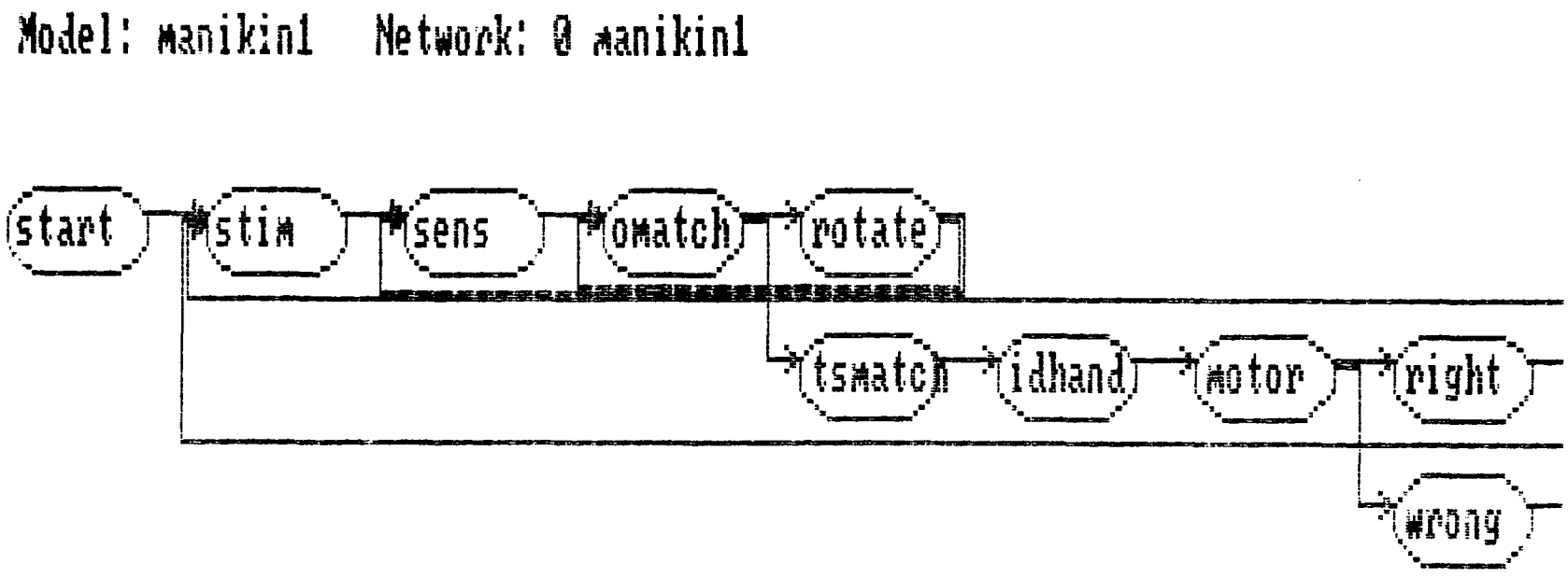


FIGURE 2

A MicroSAINT network diagram printout for the Manikin 1 model

TASK NUMBER TASK NAME

start - start

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are currently set to 0.

BEGINNING EFFECTS

n respt = number of responses, total accumulative response time and total number of mistakes are initially set to 0.

```
Task Number:  start
(1)  Name:  start                                    (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:  nrespt=0;totrespt=0;totmistk=0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
```

TASK NUMBER TASK NAME

stim stimulus

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are currently set to 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

Two things are done in the ending effect. First, a value of 1, 2, 3, or 4 is randomly selected to represent the orientation of the manikin in the stimulus and is stored in the variable called "stimulus". The function "int(4*rand()+1)" will return a value of 1, 2, 3, 4, or 5. The value of 5 is returned if rand() = 1. In this case the "if" statement following this function resets the variable stimulus to the valid value of 4.

The second thing that happens is that the time of the stimulus presentation is recorded in the variable "stimtime".

FOLLOWING TASKS

Upon the completion of this task the sensory processing task is initiated.

```
Task Number:  stim
(1)  Name:  stimulus                                  (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:  count=0;
(9)  Task's ending effect:  stimulus = int(4*rand()+1);
if stimulus == 5 then stimulus = 4;
stimtime = clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  sens          sensor  (12)  1;
(13)                        (14)
```

TASK NUMBER		TASK NAME
sens	-	sensory processing

MEAN TIME AND STANDARD DEVIATION

Smean and Ssd are variables used to define the mean time and standard deviation for this task. They are currently set to 90 and 9 respectively in the simulation scenario.

BEGINNING EFFECTS

The variables match, Merror, TSerror, IDerror, Oerror and mistake are all initialized to 0 for each execution of the model. The function of these variables within the model are explained later.

ENDING EFFECTS

None.

FOLLOWING TASKS

The match orientation task is initiated next.

```
Task Number:  sens
(1)  Name:  sensory processing                    (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Smean;
(7)  Standard deviation:  Ssd;
(8)  Task's beginning effect:  match=0;
Merror=0;
TSerror=0;
Oerror=0;
mistake=0;
IDerror=0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11) omatch        match   (12)  1;
(13)                       (14)
```

TASK NUMBER TASK NAME

omatch match orientation

MEAN TIME AND STANDARD DEVIATION

Omean and Osd are variables used to define the mean time and standard deviation for this task. They are currently set to 100 and 0 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

There are two functions which occur in the ending effect. The first is to test whether or not the orientation of the stimulus matches that of the desired internal orientation. This is done using the following expression:

"if rotate[stimulus]-count == 0 then match = 1;"

In this expression "rotate[]" is an array defined as follows:

rotate[1] = 0
rotate[2] = 1
rotate[3] = 1
rotate[4] = 2

If the value of stimulus = 1, then rotate[stimulus] = 0, count = 0, and the comparison is true so match would be set to 1. This would result in the model progressing to the target-surround match task.

If the value of stimulus is not 1, then the next task would be rotate. In addition to a time charge, rotate increments the value of count and then cycles back to this task and the test is performed again. If the value of stimulus is 2 or 3, then the model progresses to the target-surround task. If stimulus = 4 then the model will cycle through rotate again one more time.

The second thing which is done in the ending effect is to determine if an error in matching has occurred. This is done by generating a random number and comparing it to the value of the error criteria "Ocriteria". The current value of Ocriteria has been set to .97 in the simulation scenario. Each time an error occurs the variable "Oerror" is incremented to reflect multiple errors which may occur due to the cycling.

FOLLOWING TASKS

The following task will be rotate if there was no match and target-surround if the orientations matched.

```
Task Number:  omatch
(1)  Name:  match orientation                    (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Omean;
(7)  Standard deviation:  Osd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
if rotate[stimulus]-count == 0 then match = 1;
if rand() > Ocriteria then Oerror = Oerror + 1;
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  rotate        rotate  (12)  match == 0;
(13)  tsmatch       targ.   (14)  match == 1;
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

TASK NUMBER TASK NAME

rotate rotate manikin

MEAN TIME AND STANDARD DEVIATION

Rmean and Rsd are variables used to define the mean time and standard deviation for this task. They are currently set to 100 and 0 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

The ending effect of this task is to increment the variable count. This reflects the number of times the stimulus has been rotated.

FOLLOWING TASKS

There are two possible paths which may be taken out of this task. The first is to complete the rotation cycle by going back to the match orientation task. The second is to go back to the sensory processing task and begin processing the stimulus again. This branching back to the sensory processing task reflects the possibility that the subject gets confused or makes and error which he catches and then goes back to look at the stimulus again.

The branch that is taken is determined probabilistically using the variable "pcorrect". This variable reflects the probability that the subject does not get confused or make an error and is currently set to .97 in the simulation scenario. "pcorrect" is used for the branch back to the orientation match and its logical compliment, 1-pcorrect, branches to sensory processing.

```
Task Number:  rotate
(1)  Name:  rotate manakin                          (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Rmean;
(7)  Standard deviation:  Rsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  count = count + 1;
(10) Decision Type:  Probabilistic
     Following Task/Network:        Probability Of Taking
      Number:        Name:           This Path:
(11) omatch         match    (12)  pcorrect;
(13) sens           sensor   (14)  1-pcorrect;
(15)                         (16)
```

TASK NUMBER TASK NAME

tsmatch target surround match

MEAN TIME AND STANDARD DEVIATION

TSmean and TSsd are variables used to define the mean time and standard deviation for this task. They are currently set to 100 and 0 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

The function of the ending effect is to determine if an error in matching has occurred. This is done by generating a random number and comparing it to the value of the error criteria "TScriteria". The current value of TScriteria has been set to .97 in the simulation scenario. If an error occurs the variable "TSerror" is set equal to 1.

FOLLOWING TASKS

The ID hand task is initiated next.

```
Task Number:  tsmatch
(1)  Name:  targ. surround match                    (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  TSmean;
(7)  Standard deviation:  TSsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  if rand() > TScriteria then TSerror = 1;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  idhand        ID han  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

TASK NUMBER TASK NAME

idhand ID hand with object

MEAN TIME AND STANDARD DEVIATION

IDmean and IDsd are variables used to define the mean time and standard deviation for this task. They are currently set to 100 and 0 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

The function of the ending effect is to determine if an error in identification has occurred. This is done by generating a random number and comparing it to the value of the error criteria "IDcriteria". The current value of IDcriteria has been set to .97 in the simulation scenario. If an error occurs the variable "IDerror" is set equal to 1.

FOLLOWING TASKS

The motor response task is initiated next.

Task Number: idhand
(1) Name: ID hand with object (2) Type: Task
(3) Upper Network: 0 manikin1
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: IDmean;
(7) Standard deviation: IDsd;
(8) Task's beginning effect:
(9) Task's ending effect: if rand()) IDcriteria then IDerror = 1;
(10) Decision Type: Single choice

	Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11)	motor	motor	(12)	1;
(13)			(14)	
(15)			(16)	
(17)			(18)	
(19)			(20)	
(21)			(22)	
(23)			(24)	

TASK NUMBER TASK NAME

motor motor response

MEAN TIME AND STANDARD DEVIATION

Mmean and Msd are variables used to define the mean time and standard deviation for this task. They are currently set to 100 and 10 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

Three functions are performed in the ending effect for this task. First, the response time for the performance of this trail is recorded using the expression:

"resptime = clock - stimtime;"

Second, a determination of whether an error in response has occurred is made. This is done by generating a random number and comparing it to the value of the error criteria "Mcriteria". The current value of Mcriteria has been set to .97 in the simulation scenario. If an error occurs the variable "Merror" is set equal to 1.

Finally, a determination of the overall effects of the errors made in each of the steps is made. This is done in two parts. First, since the error for the orientation match increments each time an error is made in a cycle, it is possible to get values of 0, 1, 2, or 3 for the variable Oerror. Values of 0 or 2 result in a net outcome of no error since if the error occurs twice it corrects itself. Values of 1 or 3 result in a net outcome of an error. Given this, the following expression is used to set the value of Oerror to 0 or 1 (i.e., no error, error):

"if Oerror > 1 then Oerror = Oerror - 2;"

In this expression if Oerror is 0 or 1 there is no effect. If it is 2 or 3, it is changed to 0 or 1 respectively.

The second step is to determine the overall error for the process. This is done by summing up the individual errors and assigning that total to the variable Terror. Based on the same logic given in the computation of Oerror above, any time the value of Terror is even (i.e., 0, 2, or 4), the net result is a correct answer. If the value of Terror is odd (i.e., 1 or 3), then the net result is an incorrect answer. If Terror is odd the variable "mistake" is set to 1 to reflect the error.

The number of responses, mistakes and percentage right and wrong are upgraded after each trial.

FOLLOWING TASKS

One of two dummy tasks "right" or "wrong" is initiated based on the value of mistake. This is done so that separate data files may be used to record correct and incorrect trials. If mistake = 0 then the "right" task is executed and the "wrong" task is executed if mistake = 1.

```
Task Number:  motor
(1)  Name:  motor response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Mmean;
(7)  Standard deviation:  Msd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  resptime = clock -stimtime;
if rand() > Mcriteria then Merror = 1;
if Oerror > 1 then Oerror = Oerror - 2;
Terror = Oerror + TSerror + IDerror + Merror;
if Terror == 1 | Terror == 3 then mistake = 1;
nrespt=nrespt+1;totrespt=totrespt+resptime;
totmistk=totmistk+mistake;permistk=100*totmistk/nrespt;
perright=100-permistk;avrespt=totrespt/nrespt;
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  right         right   (12)  mistake == 0;
(13)  wrong         wrong   (14)  mistake == 1;
```

TASK NUMBER	TASK NAME
right	right data

MEAN TIME AND STANDARD DEVIATION

The mean and standard deviation for this task are both 0 since it is a dummy task only used for data collection.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The stimulus task is initiated next. This completes one trial or cycle of the model.

```
Task Number:  right
(1)  Name:  right response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
(13)                        (14)
```

TASK NUMBER TASK NAME

wrong wrong data

MEAN TIME AND STANDARD DEVIATION

The mean and standard deviation for this task are both 0 since it is a dummy task only used for data collection.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The stimulus task is initiated next. This completes one trial or cycle of the model.

```
Task Number:  wrong
(1)  Name:  wrong response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin1
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
(13)                        (14)
```

SIMULATION SCENARIO

```
( 1)  Event Time:          0.00
( 2)  Expression:          rotate[1] = 0;
rotate[2] = 1;
rotate[3] = 1;
rotate[4] = 2;
Smean = 90;
Ssd = 9;
Omean = 100;
Osd = 0;
Rmean = 100;
Rsd = 0;
TSmean = 100;
TSsd = 0;
IDmean = 100;
IDsd = 0;
Mmean = 100;
Msd = 10;
Mcriteria = .97;
TScriteria = .97;
IDcriteria = .97;
Ocriteria = .97;
pcorrect = .97;

( 1)  Event Time:          360000.00
( 2)  Expression:          halt();
```

Data Collection

Four snapshots files are used to collect time and error data. One file stores the variables resptime and mistake for all trails and is taken at the end of the motor response task. Two files store the variable resptime and are taken at the end of the right and wrong tasks respectively. The last file stores;

nrespt = number of responses
resptime = response time
avrespt = average response time
Amistk = accumulative number of mistakes
perright = percentage of right responses

These snapshot files are titled MANall, MANright, MANwrong, and manisum. We list here Microsaint out put of these snapshots.

SNAPSHOTS OF EXECUTION Model Name: manikin1

```
        Trigger:
( 1)    motor           motor response
( 2)    right           right response
( 3)    wrong           wrong response
( 4)    motor           motor response


( 1)  Trigger:          End trigger
( 2)  Task/Network:     motor           motor response


( 6)  Snapshot file:    MAN1all
Variables to store:
( 7)  resptime                      ( 8)  mistake
( 9)                                (10)


( 1)  Trigger:          End trigger
( 2)  Task/Network:     right           right response


( 6)  Snapshot file:    MAN1righ
Variables to store:
( 7)  resptime                      ( 8)
( 9)                                (10)


( 1)  Trigger:          End trigger
( 2)  Task/Network:     wrong           wrong response


( 6)  Snapshot file:    man1wron
Variables to store:
( 7)  resptime                      ( 8)
( 9)                                (10)


( 1)  Trigger:          End trigger
( 2)  Task/Network:     motor           motor response


( 6)  Snapshot file:    man1sum
Variables to store:
( 7)  nrespt                        ( 8)  resptime
( 9)  avrespt                       (10)  totmistk
(11)  perright                      (12)
(13)                                (14)
```

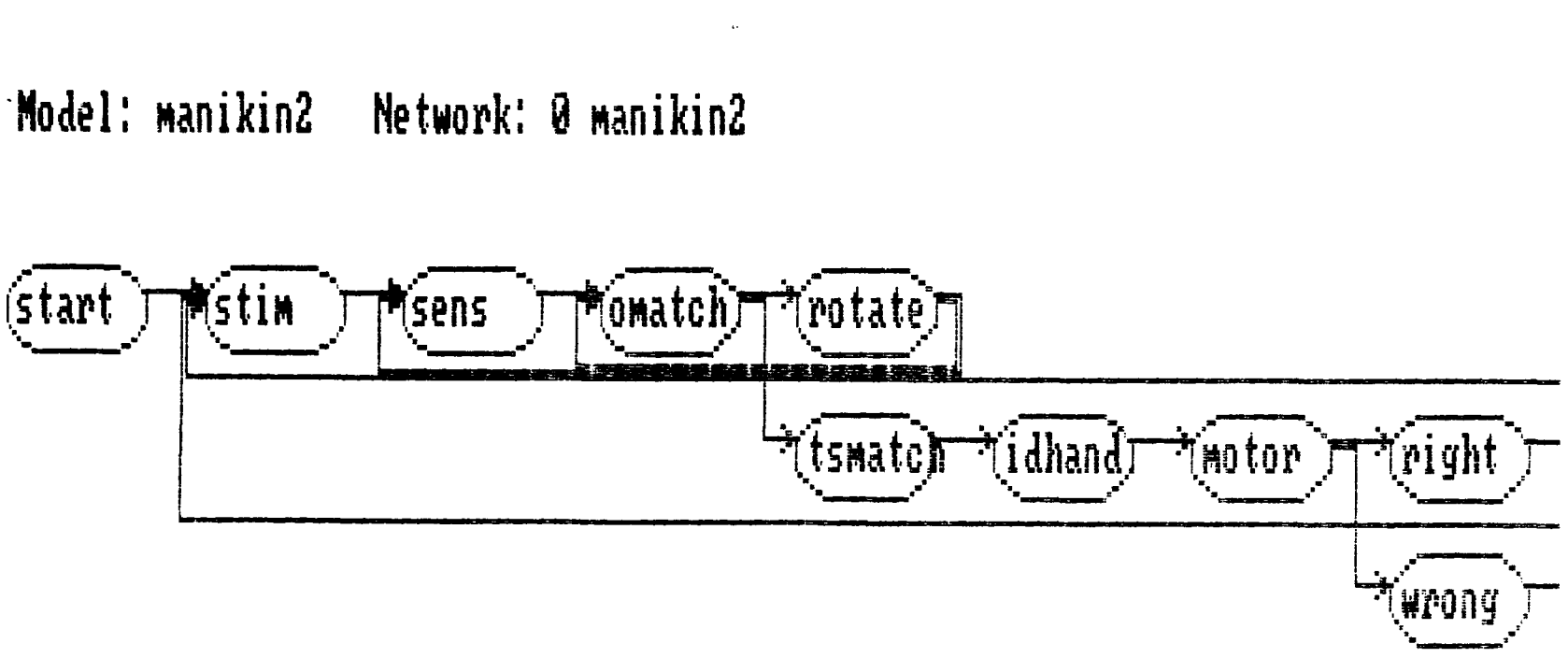


FIGURE 3

A MicroSAINT network diagram printout for the Manikin 2 model

(ii) Manikin 2 Model

Manikin 2 program is similar to Manikin 1 model except that it allows only one rotation. We list here the complete Microsaint program for Manikin 2 model.

```
TASK NETWORK

Network Number: 0
(1) Name: manikin2                                  (2) Type: Network
(3) Upper Network:
(4) Release Condition: 1;
(5) First sub-job: start  start
(6) Sub-jobs (each can be task or network):
Number:      Name:                 Type:
stim         stimulus              Task
sens         sensory processing    Task
cmatch       match orientation     Task
rotate       rotate manakin        Task
tsmatch      targ. surround match  Task
idhand       ID hand with object   Task
motor        motor response        Task
right        right response        Task
wrong        wrong response        Task
start        start                 Task


Task Number: start
(1) Name: start                                     (2) Type: Task
(3) Upper Network: 0  manikin2
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect: nrespt=0;totrespt=0;totmistk=0;
(9) Task's ending effect:
(10) Decision Type: Single choice
     Following Task/Network:      Probability Of Taking
      Number:      Name:           This Path:
(11)  stim         stimul  (12)  1;
(13)                       (14)
(15)                       (16)
(17)                       (18)
(19)                       (20)
(21)                       (22)
(23)                       (24)
```

```
Task Number:  stim
(1)  Name:  stimulus                                    (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:  count=0;
(9)  Task's ending effect:  stimulus = int(4*rand()+1);
if stimulus == 5 then stimulus = 4;
stimtime = clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  sens          sensor  (12)  1;
(13)                        (14)

Task Number:  sens
(1)  Name:  sensory processing                          (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Smean;
(7)  Standard deviation:  Ssd;
(8)  Task's beginning effect:  match=0;
Merror=0;
TSerror=0;
Oerror=0;
mistake=0;
IDerror=0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  omatch        match   (12)  1;
(13)                        (14)

Task Number:  omatch
(1)  Name:  match orientation                           (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Omean;
(7)  Standard deviation:  Osd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
if rotate[stimulus]-count == 0 then match = 1;
if rand() > Ocriteria then Oerror =Oerror+1;
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  rotate        rotate  (12)  match == 0;
(13)  tsmatch       targ.   (14)  match == 1;
(15)                        (16)
```

```
Task Number:  rotate
(1)  Name:  rotate manakin                                  (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Rmean;
(7)  Standard deviation:  Rsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  count = count + 1;
(10) Decision Type:  Probabilistic
     Following Task/Network:        Probability Of Taking
      Number:       Name:           This Path:
(11)  omatch        match    (12)  pcorrect;
(13)  sens          sensor   (14)  1-pcorrect;
(15)                         (16)
```

```
Task Number:  tsmatch
(1)  Name:  targ. surround match                            (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  TSmean;
(7)  Standard deviation:  TSsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  if rand() > TScriteria then TSerror = 1;
(10) Decision Type:  Single choice
     Following Task/Network:        Probability Of Taking
      Number:       Name:           This Path:
(11)  idhand        ID han   (12)  1;
(13)                         (14)
```

```
Task Number:  idhand
(1)  Name:  ID hand with object                             (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  IDmean;
(7)  Standard deviation:  IDsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  if rand() > IDcriteria then IDerror = 1;
(10) Decision Type:  Single choice
     Following Task/Network:        Probability Of Taking
      Number:       Name:           This Path:
(11)  motor         motor    (12)  1;
(13)                         (14)
```

```
Task Number:  motor
(1)  Name:  motor response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Mmean;
(7)  Standard deviation:  Msd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  resptime = clock -stimtime;
if rand() > Mcriteria then Merror = 1;
if Oerror > 1 then Oerror=Oerror-2;
Terror = Oerror + TSerror + IDerror + Merror;
if Terror == 1 | Terror == 3 then mistake = 1;
nrespt=nrespt+1;totrespt=totrespt+resptime;
totmistk=totmistk+mistake;permistk=100*totmistk/nrespt;
perright=100-permistk;avrespt=totrespt/nrespt;
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  right         right   (12)  mistake == 0;
(13)  wrong         wrong   (14)  mistake == 1;
(15)                        (16)
```

```
Task Number:  right
(1)  Name:  right response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
(13)                        (14)
```

```
Task Number:  wrong
(1)  Name:  wrong response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin2
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
(13)                        (14)
```

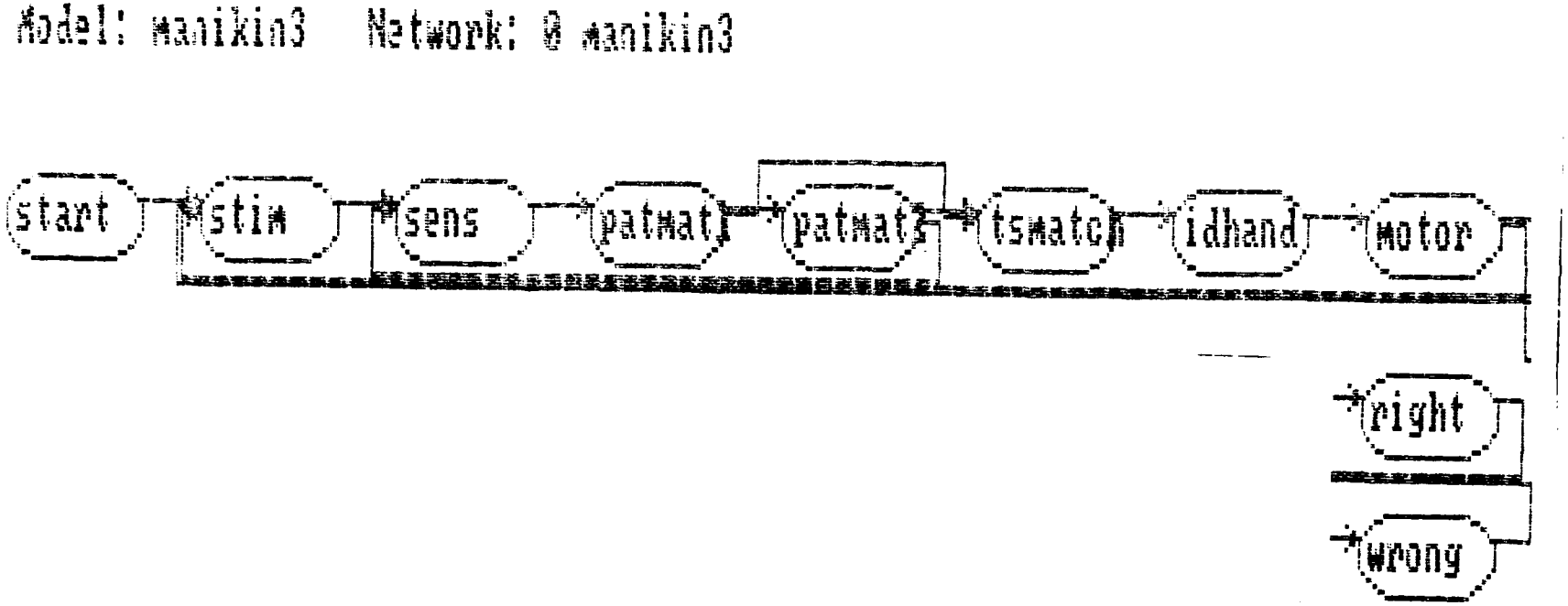


FIGURE 4

A MicroSAINT network diagram printout for the Manikin 3 model

(iii) Manikin 3 Model - No Rotation

Manikin 3 model is similar to Manikin 1 model except that the two tasks omatch (match orientation) and rotate (rotate manikin) are replaced by the two tasks patmat1 (pattern match 1) and patmat2 (pattern match 2). We list here the complete program for Manikin 3 model.

```
        TASK  NETWORK

Network Number:  0
(1)  Name:  manikin3                              (2)  Type:  Network
(3)  Upper Network:
(4)  Release Condition:  1;
(5)  First sub-job:  start  start
(6)  Sub-jobs (each can be task or network):
Number:        Name:                  Type:
stim           stimulus               Task
sens           sensory processing     Task
patmat1        patmathc1              Task
patmat2        patmatch2              Task
tsmatch        targ. surround match   Task
idhand         ID hand with object    Task
motor          motor response         Task
right          right response         Task
wrong          wrong response         Task
start          start                  Task


Task Number:  stim
(1)  Name:  stimulus                              (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:  count=0;
(9)  Task's ending effect:  stimulus = int(4*rand()+1);
if stimulus == 5 then stimulus = 4;
stimtime = clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  sens          sensor  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

```
Task Number:  sens
(1)  Name:  sensory processing                    (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Smean;
(7)  Standard deviation:  Ssd;
(8)  Task's beginning effect:  match=0;
Merror=0;
TSerror=0;
pterror1=0;
mistake=0;
pterror2=0;
IDerror=0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  patmat1       patmat  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

```
Task Number:  patmat1
(1)  Name:  patmathc1                             (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  patmean1;
(7)  Standard deviation:  patsd1;
(8)  Task's beginning effect:
(9)  Task's ending effect:
if rand() > ptcrit1 then pterror1=pterror1+1;
if ptrgn[stimulus]-count==0 then match=1;
count=count+1;
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  patmat2       patmat  (12)  match==0;
(13)  tsmatch       targ.   (14)  match==1;
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

```
Task Number:  patmat2
(1)  Name:  patmatch2                                   (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  patmean2;
(7)  Standard deviation:  patsd2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  if rand() > ptcrit2 then pterror2=1;
(10) Decision Type:  Probabilistic
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  tsmatch       targ.   (12)  pcorrect;
(13)  sens          sensor  (14)  1-pcorrect;
```

```
Task Number:  tsmatch
(1)  Name:  targ. surround match                        (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  TSmean;
(7)  Standard deviation:  TSsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  if rand() > TScriteria then TSerror = 1;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  idhand        ID han  (12)  1;
(13)                        (14)
```

```
Task Number:  idhand
(1)  Name:  ID hand with object                         (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  IDmean;
(7)  Standard deviation:  IDsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  if rand() > IDcriteria then IDerror = 1;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  motor         motor   (12)  1;
(13)                        (14)
```

```
Task Number:  motor
(1)  Name:  motor response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Mmean;
(7)  Standard deviation:  Msd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  resptime = clock -stimtime;
if rand() > Mcriteria then Merror = 1;
if pterror1 > 1 then pterror1=pterror1-2;
Terror = pterror1+pterror2+ TSerror + IDerror + Merror;
if Terror==5 then Terror=3;
if Terror == 1 | Terror == 3  then mistake = 1;
nrespt=nrespt+1;totrespt=totrespt+resptime;
totmistk=totmistk+mistake;permistk=100*totmistk/nrespt;
perright=100-permistk;avrespt=totrespt/nrespt;
(10) Decision Type:  Tactical
     Following Task/Network:       Tactical Expression:
      Number:       Name:
(11)  right         right    (12)  mistake == 0;
(13)  wrong         wrong    (14)  mistake == 1;
(15)                         (16)
(17)                         (18)
(19)                         (20)
(21)                         (22)
(23)                         (24)
```

```
Task Number:  right
(1)  Name:  right response                          (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul   (12)  1;
(13)                         (14)
(15)                         (16)
(17)                         (18)
(19)                         (20)
(21)                         (22)
(23)                         (24)
```

```
Task Number:  wrong
(1)  Name:  wrong response                                  (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

```
Task Number:  start
(1)  Name:  start                                           (2)  Type:  Task
(3)  Upper Network:  0  manikin3
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:  nrespt=0;totrespt=0;totmistk=0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          stimul  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

```
                SIMULATION SCENARIO     Model Name:  manikin1

            Time:       Expression:
      ( 1)  0.00        rotate[1] = 0; MORE
      ( 2)  360000.00   halt();
```

```
( 1)  rotate[1] = 0;
( 2)  rotate[2] = 1;
( 3)  rotate[3] = 1;
( 4)  rotate[4] = 2;
( 5)  Smean = 90;
( 6)  Ssd. = 9;
( 7)  Umean = 100;
( 8)  Usd = 0;
( 9)  Rmean = 100;
(10)  Rsd = 0;
(11)  TSmean = 100;
(12)  TSsd = 0;
(13)  IDmean = 100;
(14)  IDsd = 0;
(15)  Mmean = 100;
(16)  Msd = 10;
```

(iv) Sample outputs from Manikin 1,2, and 3 Models

The following table represents a sample output from the three programs. Test time was set at 6 minutes. We included results from the first and last five trials. The results indicate that subjects with strategy 1 (maximum of two rotations), strategy 2 (maximum of one rotation), and strategy 3 (no rotation but matching two preferred positions) will perform 519, 560, and 666 trials in the allotted time of 6 minutes respectively. The average response time per trial is 693 MS, 642 MS, and 540 MS and the percentage of correct answers (identification) are 86.5%, 88.2% and 85.9%.

TABLE 4

Model: manikin1

```
"clock" "nrespt" "resptime" "avrespt" "totmistk" "perright" "Trigger:" "Jobs
485.493774 1 485.493774 485.493774 0 100.000000 "End" "motor"
1189.873047 2 704.379272 594.936523 0 100.000000 "End" "motor"
2060.314941 3 870.441895 686.771667 0 100.000000 "End" "motor"
2941.528564 4 881.213623 735.382141 0 100.000000 "End" "motor"
3650.687988 5 709.159424 730.137573 0 100.000000 "End" "motor"

356389.125000 515 492.843750 692.017700 69 86.601944 "End" "motor"
357354.250000 516 965.125000 692.546997 70 86.434113 "End" "motor"
358267.906250 517 913.656250 692.974670 70 86.460350 "End" "motor"
359145.468750 518 877.562500 693.330994 70 86.486488 "End" "motor"
359640.968750 519 495.500000 692.949829 70 86.512527 "End" "motor"
```

Model: manikin2

```
"clock" "nrespt" "resptime" "avrespt" "totmistk" "perright" "Trigger:" "Jobs
485.493774 1 485.493774 485.493774 0 100.000000 "End" "motor"
1189.873047 2 704.379272 594.936523 0 100.000000 "End" "motor"
1872.865845 3 682.992798 624.288635 0 100.000000 "End" "motor"
2560.882080 4 688.016235 640.220520 0 100.000000 "End" "motor"

357430.531250 556 679.781250 642.860657 65 88.309357 "End" "motor"
357904.625000 557 474.093750 642.557678 65 88.330338 "End" "motor"
358372.562500 558 467.937500 642.244751 65 88.351257 "End" "motor"
359028.218750 559 655.656250 642.268738 66 88.193199 "End" "motor"
359742.093750 560 713.875000 642.396606 66 88.214287 "End" "motor"
```

Model: manikin3

```
"clock" "nrespt" "resptime" "avrespt" "totmistk" "perright" "Trigger:" "Jobs
485.493774 1 485.493774 485.493774 0 100.000000 "End" "motor"
986.496704 2 501.002930 493.248352 0 100.000000 "End" "motor"
1479.086304 3 492.589600 493.028778 0 100.000000 "End" "motor"
2060.100342 4 581.014038 515.025085 0 100.000000 "End" "motor"
2639.900146 5 579.799805 527.980042 0 100.000000 "End" "motor"

357757.125000 662 508.937500 540.418640 92 86.102722 "End" "motor"
358225.031250 663 467.906250 540.309265 92 86.123680 "End" "motor"
358794.750000 664 569.718750 540.353516 93 85.993973 "End" "motor"
359295.875000 665 501.125000 540.294556 93 86.015038 "End" "motor"
359899.437500 666 603.562500 540.389526 94 85.885887 "End" "motor"
```

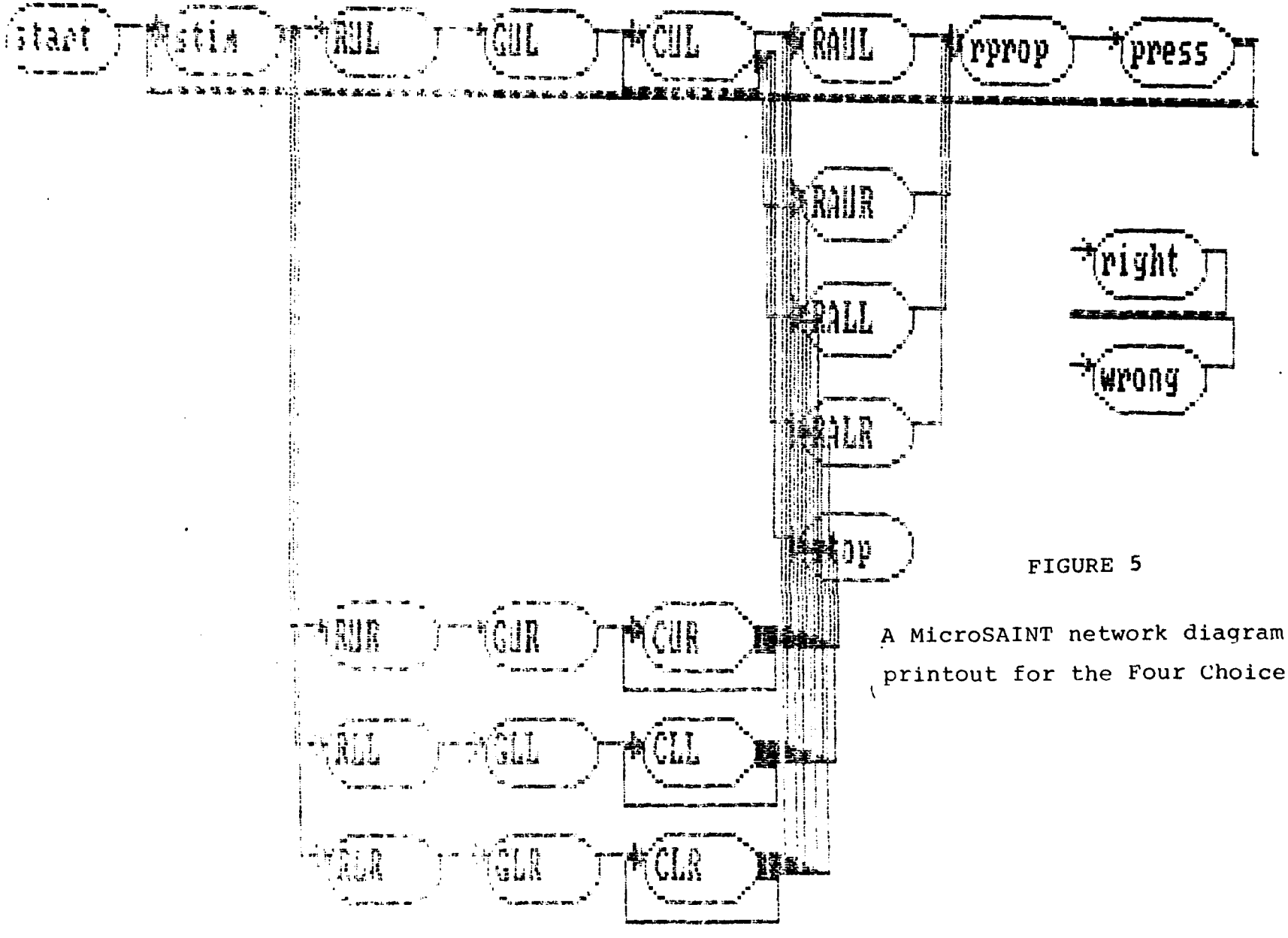


FIGURE 5

A MicroSAINT network diagram printout for the Four Choice model

III. THE FOUR CHOICE REACTION TIME MODEL

Introduction

The basic representation selected for modeling this test is a neuroanatomical one. The model reflects the major neuroanatomical structures and phenomena involved in processing and responding to the information presented in the test. The basic flow of the model is as follows. The stimulus is presented and propagated through the retinal ganglion and the geniculate to the cortex. In the cortex, the effects of the stimulus build up to a specified threshold for detection. Once this threshold is exceeded, an association between the location of the stimulus and the appropriate motor response is made. The signal to respond is then sent through the nervous system and the response is made.

In addition to the neuroanatomical flow, the model also reflects the nature of the task and its impact on perception and processing. The stimulus may be presented in one of four quadrants: upper left (UL), upper right (UR), lower left (LL) or lower right (LR). In order to reflect this, the model assumes that each of these four quadrants are processed via separate channels. It is also assumed that stimulation in one channel will also result in a lower level of stimulation in the other channels. These lower levels of stimulation may, from time to time, cause their respective channels to indicate the presence of the stimulus. This results in an error for the process.

Errors may also occur as a result of inappropriate response associations. That is, while the correct channel may have been stimulated, the association of that channel with the correct motor response may be in error. This phenomena is reflected in the model.

Organization for the Documentation

This document is a mix of model printouts from Micro SAINT and word-processed explanations. It begins with the network diagram and the listing of the tasks. It then proceeds to the individual task descriptions. This is generally a page for page mix of Micro SAINT printouts and explanations. At the end of this are the Micro SAINT printouts of the Simulation Scenario, Snapshots, Functions, and Continuous Variable Changes as appropriate. These are referenced in the explanations.

Data Collection

Two basic sets of snapshots are used to collect the data for the model. The first set consists of three files for time and error data. One file stores the variables resptime, channel, output, and match for all trails and is taken at the end of the press button task. The other two store the variables resptime, channel, and output and are taken at the end of the right and wrong tasks respectively. These snapshot files are titled C4all, C4right, and C4wrong.

The second set of snapshots are used to record the behavior of the growth functions in the cortex tasks. Since there are four cortex tasks there are four snapshot files, one for each task. In these files the values for Ex and k for the task (e.g., ExUR and kUR) are recorded each time the task executes. These snapshot files are titled C4ExUL, C4ExUR, C4ExLL, and C4ExLR.

Task Descriptions

In this section an explanation of each of the tasks in the model is provided. In these explanations, the variables, equations, and logic used in the task are described as well as the relationship of the task to other tasks in the model. In this particular model there are a number of very similar tasks since four parallel channels are being modeled. These parallel tasks are identical except for changes in variable names and following tasks. As a result, task descriptions are provided for the tasks in only one of the channels with the parallel tasks being listed with that description.

All times are in milliseconds. The function rand() returns a random number from a uniform distribution. Whenever the character pairs UL, UR, LL, or LR appear as parts of variable or task names they represent, respectively, the upper left, upper right, lower left, and lower right quadrants or channels.

```
        TASK  NETWORK

Network Number:  0
(1)  Name:  4choice                                  (2)  Type:  Network
(3)  Upper Network:
(4)  Release Condition:  1;
(5)  First sub-job:  start  start
(6)  Sub-jobs (each can be task or network):
Number:        Name:                  Type:
stim           present stimulus       Task
RUL            retinal ganglion       Task
RUR            retinal ganglion       Task
RLL            retinal ganglion       Task
RLR            retinal ganglion       Task
GUL            geniculate             Task
GUR            geniculate             Task
GLL            geniculate             Task
GLR            geniculate             Task
CUL            cortex                 Task
CUR            cortex                 Task
CLL            cortex                 Task
CLR            cortex                 Task
RAUL           response association   Task
RAUR           response association   Task
RALL           response association   Task
RALR           response association   Task
rprop          response propogation   Task
press          press button           Task
right          right data             Task
wrong          wrong data             Task
stop           stop dummy             Task
start          start                  Task
```

In the start task, we initialize nrespt = number of responses, totwrong = accumulative number of wrong responses, and totrespt = total response time.

```
Task Number:  start
(1)  Name:  start                                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:  nrespt=0;totwrong=0;totrespt=0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          presen  (12)  1;
(13)                        (14)
```

TASK NUMBER TASK NAME

stim present stimulus

MEAN TIME AND STANDARD DEVIATION

Smean and Ssd are variables used to define the mean time and standard deviation for this task. They are currently set to 0 in the simulation scenario.

BEGINNING EFFECTS
The variables processedUL, processedUR, processedLL, processedLR, allstop, and match are all initialized to 0 for each execution of the model. The function of these variables within the model are explained later.

ENDING EFFECTS
Two things are done in the ending effect. First, a value of 0, 1, 2, or 3 is randomly selected for the variable channel. This identifies the channel, or quadrant in which the stimulus is presented (0 = UL, 1 = UR, 2 = LL, 3 = LR). The function "int(4*rand())" will return a value of 0, 1, 2, 3, or 4. The value of 4 is returned if rand() = 1. In this case the "if" statement following this function resets the variable channel to the valid value of 3.

The second thing that happens is that the time of the stimulus presentation is recorded in the variable "stimtime".

FOLLOWING TASKS
Upon the completion of this task all four of the channels are started with the initiation of the retinal ganglion tasks reflecting each quadrant.

```
Task Number: stim
(1) Name: present stimulus                    (2) Type: Task
(3) Upper Network: 0 4choice
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: Smean;
(7) Standard deviation: Ssd;
(8) Task's beginning effect: processedUL = 0;
processedUR = 0;
processedLL = 0;
processedLR = 0;
allstop = 0;
match = 0;
(9) Task's ending effect: channel = int(4*rand());
if channel == 4 then channel = 3;
stimtime = clock;
(10) Decision Type: Multiple
     Following Task/Network:      Probability Of Taking
      Number:      Name:           This Path:
(11) RUL          retina (12) 1;
(13) RUR          retina (14) 1;
(15) RLL          retina (16) 1;
(17) RLR          retina (18) 1;
```

TASK NUMBER	TASK NAME
RUL	retinal ganglion
RUR	retinal ganglion
RLL	retinal ganglion
RLR	retinal ganglion

MEAN TIME AND STANDARD DEVIATION

RGmean and RGsd are variables used to define the mean time and standard deviation for this task. They are currently set to 45 and 5 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The geniculate task for the channel is initiated next.

Task Number: RUL
(1) Name: retinal ganglion *(2) Type: Task*
(3) Upper Network: 0 4choice
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: RGmean;
(7) Standard deviation: RGsd;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

	Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11)	GUL	genicu	(12)	1;
(13)			(14)	
(15)			(16)	
(17)			(18)	
(19)			(20)	
(21)			(22)	
(23)			(24)	

```
Task Number:  RUR
(1)  Name:  retinal ganglion                          (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RGmean;
(7)  Standard deviation:  RGsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:        Name:          This Path:
(11)  GUR            genicu  (12)  1;
(13)                         (14)
```

```
Task Number:  RLL
(1)  Name:  retinal ganglion                          (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RGmean;
(7)  Standard deviation:  RGsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:        Name:          This Path:
(11)  GLL            genicu  (12)  1;
(13)                         (14)
```

```
Task Number:  RLR
(1)  Name:  retinal ganglion                          (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RGmean;
(7)  Standard deviation:  RGsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:        Name:          This Path:
(11)  GLR            genicu  (12)  1;
(13)                         (14)
```

TASK NUMBER	TASK NAME
GUL	geniculate
GUR	geniculate
GLL	geniculate
GLR	geniculate

MEAN TIME AND STANDARD DEVIATION

GENmean and GENsd are variables used to define the mean time and standard deviation for this task. They are currently set to 45 and 5 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The cortex task for the channel is initiated next.

Task Number: GUL
(1) Name: geniculate *(2) Type: Task*
(3) Upper Network: 0 4choice
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: GENmean;
(7) Standard deviation: GENsd;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:	Probability Of Taking This Path:
(11) CUL	cortex (12)	1;
(13)	(14)	
(15)	(16)	
(17)	(18)	
(19)	(20)	
(21)	(22)	
(23)	(24)	

```
Task Number:  GUR
(1)  Name:  geniculate                                  (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  GENmean;
(7)  Standard deviation:  GENsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  CUR           cortex  (12)  1;
(13)                        (14)
```

```
Task Number:  GLL
(1)  Name:  geniculate                                  (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  GENmean;
(7)  Standard deviation:  GENsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  CLL           cortex  (12)  1;
(13)                        (14)
```

```
Task Number:  GLR
(1)  Name:  geniculate                                  (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  GENmean;
(7)  Standard deviation:  GENsd;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  CLR           cortex  (12)  1;
(13)                        (14)
```

TASK NUMBER	TASK NAME
CUL	cortex
CUR	cortex
CLL	cortex
CLR	cortex

MEAN TIME AND STANDARD DEVIATION

Cmean and Csd are variables used to define the mean time and standard deviation for this task. They are currently set to 50 and 0 respectively in the simulation scenario.

BEGINNING EFFECTS

The purpose of the beginning effect is to compute the level of excitation for the channel and determine if it has exceeded the threshold for executing the motor response. The function of each of the expressions is as follows:

"if channel == 0 then kUL = 1 else kUL = .3;"

This expression tests whether or not this channel has been selected (i.e., channel == 0 for UL, 1 for UR, 2 for LL, 3 for LR) and then sets the value of k to 1 if it is the selected channel and to .3 if it is not. k is a parameter of the growth function for the channel excitation. The larger it is the faster the function grows.

"ExUL = kUL*(1 - (2.71828^(-1*(clock-stimtime)/tau))) + normal(0,0.1);"

This expression returns the value for the growth function for this channel. It is a negative exponential function based on the elapsed time since the presentation of the stimulus (i.e., clock-stimtime). The variable "tau" determines the shape of the growth function and is currently set to 200 in the simulation scenario. The function "normal(0,0.1)" returns a value from a normal distribution with a mean of 0 and a standard deviation of 0.1. It reflects the error for the expression. This allows for the possibility of a channel which is not selected to become stimulated in spite of its lower growth rate.

"if ExUL >= threshhold then processedUL = 1;"

This expression tests the computed value of Ex for the channel to see if it has exceeded the threshold for stimulation. If it has, the flag "processed" for the channel is set to 1 to indicate that this channel was stimulated. The value of threshhold is currently set to .8 in the simulation scenario.

"if ExUL >= threshhold then allstop = 1;"

This expression makes the same test as the previous one but sets the "allstop" flag to 1. This causes the other three channels to stop processing.

ENDING EFFECTS

The ending effect of this task is to set the variable "associate" for the channel equal to a random number. This variable is used to reflect errors in matching the stimulus to the response. That is, the correct channel may be identified, but the subject may still press the incorrect key due to a mismatch of the correct motor response.

FOLLOWING TASKS

There a six mutually exclusive following tasks. One of the first four will be taken if the growth function for the channel exceeds the threshold. This is reflected by the value of the "processed" flag being equal to 1. The expressions for determining which of these four channels to take are as follows:

Following Task	Expression
RAUL	processedUL == 1 & associateUL <= .99;
RAUR	processedUL == 1 & associateUL > .99 & associateUL <= .993;
RALL	processedUL == 1 & associateUL > .993 & associateUL <= .996;
RALR	processedUL == 1 & associateUL > .996;

In the ending effect of the task the variable associate was set to a random number. This is tested against values for each of the possible matches to a motor response. The most likely (.99) path is the correct response. There is the possibility (.003) that each of the other paths might be taken if an incorrect match is made (i.e., associated > .99). While the determination of whether or not the match occurs is performed here, the time charge for this is reflected in the next task (i.e., RAUL, RAUR, etc.).

The fifth path will be taken if the threshold has not been exceeded by this channel or any other channel. This is reflected by testing the flags for this task being "processed" or the flag for another task (allstop) being processed. The path simply executes this task again to compute a new excitation value to test against the threshold.

The sixth and final path is taken if the "allstop" flag has been set. This indicates that another channel exceeded the threshold.

This path leads to a dummy task called stop which performs no function other than as a following task to end the cycling of the cortex tasks.

```
Task Number:  CUL
(1)  Name:  cortex                                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Cmean;
(7)  Standard deviation:  Csd;
(8)  Task's beginning effect:
if channel == 0 then kUL = 1 else kUL = .3;
ExUL = kUL*(1 - (2.71828^(-1*(clock-stimtime)/tau))) + normal(0,0.1);
if ExUL >= threshhold then processedUL = 1;
if ExUL >= threshhold then allstop = 1;
(9)  Task's ending effect:  associateUL = rand();
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  RAUL          respon  (12)
processedUL == 1 & associateUL <= .99;
(13)  RAUR          respon  (14)
processedUL == 1 & associateUL > .99 & associateUL <= .993;
(15)  RALL          respon  (16)
processedUL == 1 & associateUL > .993 & associateUL <= .996;
(17)  RALR          respon  (18)
processedUL == 1 & associateUL > .996;
(19)  CUL           cortex  (20)
processedUL == 0 & allstop == 0;
(21)  stop          stop d  (22)  allstop == 1;
(23)                        (24)
```

```
Task Number:  CUR
(1)  Name:  cortex                                   (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Cmean;
(7)  Standard deviation:  Csd;
(8)  Task's beginning effect:
if channel == 1 then kUR = 1 else kUR = .3;
ExUR = kUR*(1 - (2.71828^(-1*(clock-stimtime)/tau))) + normal(0,0.1);
if ExUR >= threshhold then processedUR = 1;
if ExUR >= threshhold then allstop = 1;
(9)  Task's ending effect:  associateUR = rand();
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  RAUR          respon  (12)
processedUR == 1 & associateUR <= .99;
(13)  RAUL          respon  (14)
processedUR == 1 & associateUR > .99 & associateUR <= .993;
(15)  RALL          respon  (16)
processedUR == 1 & associateUR > .993 & associateUR <= .996;
(17)  RALR          respon  (18)
processedUR == 1 & associateUR > .996;
(19)  CUR           cortex  (20)
processedUR == 0 & allstop == 0;
(21)  stop          stop d  (22)  allstop == 1;

Task Number:  CLL
(1)  Name:  cortex                                   (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Cmean;
(7)  Standard deviation:  Csd;
(8)  Task's beginning effect:
if channel == 2 then kLL = 1 else kLL = .3;
ExLL = kLL*(1 - (2.71828^(-1*(clock-stimtime)/tau))) + normal(0,0.1);
if ExLL >= threshhold then processedLL = 1;
if ExLL >= threshhold then allstop = 1;
(9)  Task's ending effect:  associateLL = rand();
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  RALL          respon  (12)
processedLL == 1 & associateLL <= .99;
(13)  RAUR          respon  (14)
processedLL == 1 & associateLL > .99 & associateLL <= .993;
(15)  RAUL          respon  (16)
processedLL == 1 & associateLL > .993 & associateLL <= .996;
(17)  RALR          respon  (18)
processedLL == 1 & associateLL > .996;
(19)  CLL           cortex  (20)
processedLL == 0 & allstop == 0;
(21)  stop          stop d  (22)  allstop == 1;
```

```
Task Number:  CLR
(1)  Name:  cortex                                   (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Cmean;
(7)  Standard deviation:  Csd;
(8)  Task's beginning effect:
if channel == 3 then kLR = 1 else kLR = .3;
ExLR = kLR*(1 - (2.71828^(-1*(clock-stimtime)/tau))) + normal(0,0.1);
if ExLR >= threshhold then processedLR = 1;
if ExLR >= threshhold then allstop = 1;
(9)  Task's ending effect:  associateLR = rand();
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  RALR          respon  (12)
processedLR == 1 & associateLR <= .99;
(13). RAUR          respon  (14)
processedLR == 1 & associateLR > .99 & associateLR <= .993;
(15)  RALL          respon  (16)
processedLR == 1 & associateLR > .993 & associateLR <= .996;
(17)  RAUL          respon  (18)
processedLR == 1 & associateLR > .996;
(19)  CLR           cortex  (20)
processedLR == 0 & allstop == 0;
(21)  stop          stop d  (22)  allstop == 1;
(23)                        (24)
```

TASK NUMBER	TASK NAME
RAUL	response association
RAUR	response association
RALL	response association
RALR	response association

MEAN TIME AND STANDARD DEVIATION

RAmean and RAsd are variables used to define the mean time and standard deviation for this task. They are currently set to 40 and 4 respectively in the simulation scenario.

BEGINNING EFFECTS

The value of the variable "output" is set equal to the number representing the selected output channel (i.e., UL = 0, UR = 1, LL = 2, LR = 3). This will be used later to identify whether the input and output channels were matched correctly.

ENDING EFFECTS

None.

FOLLOWING TASKS

The response propagation task is initiated next.

```
Task Number:  RAUL
(1)  Name:  response association                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RAmean;
(7)  Standard deviation:  RAsd;
(8)  Task's beginning effect:  output = 0;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  rprop         respon  (12)  1;
(13)                        (14)
Task Number:  RAUR
(1)  Name:  response association                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RAmean;
(7)  Standard deviation:  RAsd;
(8)  Task's beginning effect:  output = 1;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  rprop         respon  (12)  1;
(13)                        (14)
Task Number:  RALL
(1)  Name:  response association                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RAmean;
(7)  Standard deviation:  RAsd;
(8)  Task's beginning effect:  output = 2;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  rprop         respon  (12)  1;
(13)                        (14)
Task Number:  RALR
(1)  Name:  response association                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  RAmean;
(7)  Standard deviation:  RAsd;
(8)  Task's beginning effect:  output = 3;
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:       Probability Of Taking
      Number:       Name:           This Path:
(11)  rprop         respon  (12)  1;
(13)                        (14)
```

TASK NUMBER TASK NAME

rprop response propagation

MEAN TIME AND STANDARD DEVIATION

RPmean and RPsd are variables used to define the mean time and standard deviation for this task. They are currently set to 40 and 4 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The press button task is initiated next.

Task Number: rprop
(1) Name: response propogation *(2) Type: Task*
(3) Upper Network: 0 4choice
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: RPmean;
(7) Standard deviation: RPsd;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) press	press	*(12)*	*1;*
(13)		*(14)*	
(15)		*(16)*	
(17)		*(18)*	
(19)		*(20)*	
(21)		*(22)*	
(23)		*(24)*	

TASK NUMBER TASK NAME

press press button

MEAN TIME AND STANDARD DEVIATION

Pmean and Psd are variables used to define the mean time and standard deviation for this task. They are currently set to 20 and 2 respectively in the simulation scenario.

BEGINNING EFFECTS

None.

ENDING EFFECTS

Two functions are performed in the ending effect for this task. First, the response time for the performance of this trail is recorded using the expression:

"resptime = clock - stimtime;"

Second, the values in "channel" and "output" are tested against each other to determine if they match. This reflects whether or not an error was made. The value of "match" is set to 1 if they are not equal.

Third, the variables nrespt, totwrong, totrespt, avrespt, perwrong, and perright are updated after every trial.

FOLLOWING TASKS

One of two dummy tasks "right" or "wrong" is initiated based on the value of match. This is done so that separate data files may be used to record correct and incorrect trials. If match = 0 then the "right" task is executed and the "wrong" task is executed if match = 1.

```
Task Number:  press
(1)  Name:  press button                                  (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  Pmean;
(7)  Standard deviation:  Psd;
(8)  Task's beginning effect:
(9)  Task's ending effect:  resptime = clock - stimtime;
if channel <> output then match = 1;
nrespt=nrespt+1;totwrong=totwrong+match;
totrespt=totrespt+resptime;avrespt=totrespt/nrespt;
perwrong=100*totwrong/nrespt;perright=100-perwrong;
(10) Decision Type:  Tactical
     Following Task/Network:      Tactical Expression:
      Number:       Name:
(11)  right         right   (12)  match == 0;
(13)  wrong         wrong   (14)  match == 1;
(15)                        (16)
```

```
Task Number:  right
(1)  Name:  right data                                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          presen  (12)  1;
(13)                        (14)
```

```
Task Number:  wrong
(1)  Name:  wrong data                                    (2)  Type:  Task
(3)  Upper Network:  0  4choice
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stim          presen  (12)  1;
(13)                        (14)
```

SNAPSHOTS OF EXECUTION

```
( 1)  Trigger:          End trigger
( 2)  Task/Network:     press          press button
( 6)  Snapshot file:    C4all
Variables to store:
( 7)  resptime                    ( 8)  channel
( 9)  output                      (10)  match
(11)                              (12)
(13)                              (14)
(15)                              (16)
```

```
( 1)  Trigger:          End trigger
( 2)  Task/Network:     right          right data
( 6)  Snapshot file:    C4right
Variables to store:
( 7)  resptime                    ( 8)  channel
( 9)                              (10)
(11)                              (12)
(13)                              (14)
(15)                              (16)
```

```
( 1)  Trigger:          End trigger
( 2)  Task/Network:     wrong          wrong data
( 6)  Snapshot file:    C4wrong
Variables to store:
( 7)  resptime                    ( 8)  channel
( 9)  output                      (10)
(11)                              (12)
```

```
( 1)  Trigger:          End trigger
( 2)  Task/Network:     press          press button
```

```
( 6)  Snapshot file:    4csum
Variables to store:
( 7)  nrespt                      ( 8)  resptime
( 9)  avrespt                     (10)  totwrong
(11)  perright                    (12)
(13)                              (14)
(15)                              (16)
```

```
( 1)  Trigger:          End trigger
( 2)  Task/Network:     CLR            cortex
( 6)  Snapshot file:    C4ExLR
Variables to store:
( 7)  ExLR                        ( 8)  kLR
( 9)                              (10)
```

```
( 1)  Trigger:              End trigger
( 2)  Task/Network:         CLL                cortex
( 6)  Snapshot file:        C4ExLL
Variables to store:
```

```
( 1)  Trigger:              End trigger
( 2)  Task/Network:         CUL                cortex
( 6)  Snapshot file:        C4ExUL
Variables to store:
( 7)  ExUL                                ( 8)  kUL
( 9)                                      (10)
(11)                                      (12)
(13)                                      (14)
(15)                                      (16)
```

```
( 1)  Trigger:              End trigger
( 2)  Task/Network:         CUR                cortex
( 6)  Snapshot file:        C4ExUR
Variables to store:
( 7)  ExUR                                ( 8)  kUR
( 9)                                      (10)
```

SIMULATION SCENARIO

```
( 1)  Event Time:           0.00
( 2)  Expression:           RGmean = 45;
RGsd = 5;
GENmean = 45;
GENsd = 5;
Cmean = 50;
Csd = 0;
RAmean = 40;
RAsd = 4;
RPmean = 40;
RPsd = 4;
Pmean = 20;
Psd = 2;
Smean = 0;
Ssd = 0;
tau = 200;
threshhold = 0.8;
```

```
( 1)  Event Time:           360000.00
( 2)  Expression:           halt();
```

Sample Output From 4 Choice Model

The following table represents a sample output from the 4 choice model. Test time was set at 6 minutes. We included results from the first and last five trials. The total number of trials is 847. The average response time and the percentage right are 424.8 MS and 100%, respectively.

TABLE 5

Model: 4choice

"clock" "nrespt" "resptime" "avresp:" "totwrong" "perright" "Trigger:" "Job:
445.098450 1 445.098450 445.098450 0 100.000000 "End" "press"
938.958191 2 493.859741 469.479095 0 100.000000 "End" "press"
1379.381226 3 440.423035 459.793732 0 100.000000 "End" "press"
1818.765991 4 439.384766 454.691498 0 100.000000 "End" "press"
2261.994141 5 443.228149 452.398834 0 100.000000 "End" "press"

358164.250000 843 386.375000 424.868622 0 100.000000 "End" "press"
358536.437500 844 372.187500 424.806213 0 100.000000 "End" "press"
358963.250000 845 426.812500 424.808594 0 100.000000 "End" "press"
359362.500000 846 399.250000 424.778381 0 100.000000 "End" "press"
359810.625000 847 448.125000 424.805939 0 100.000000 "End" "press"

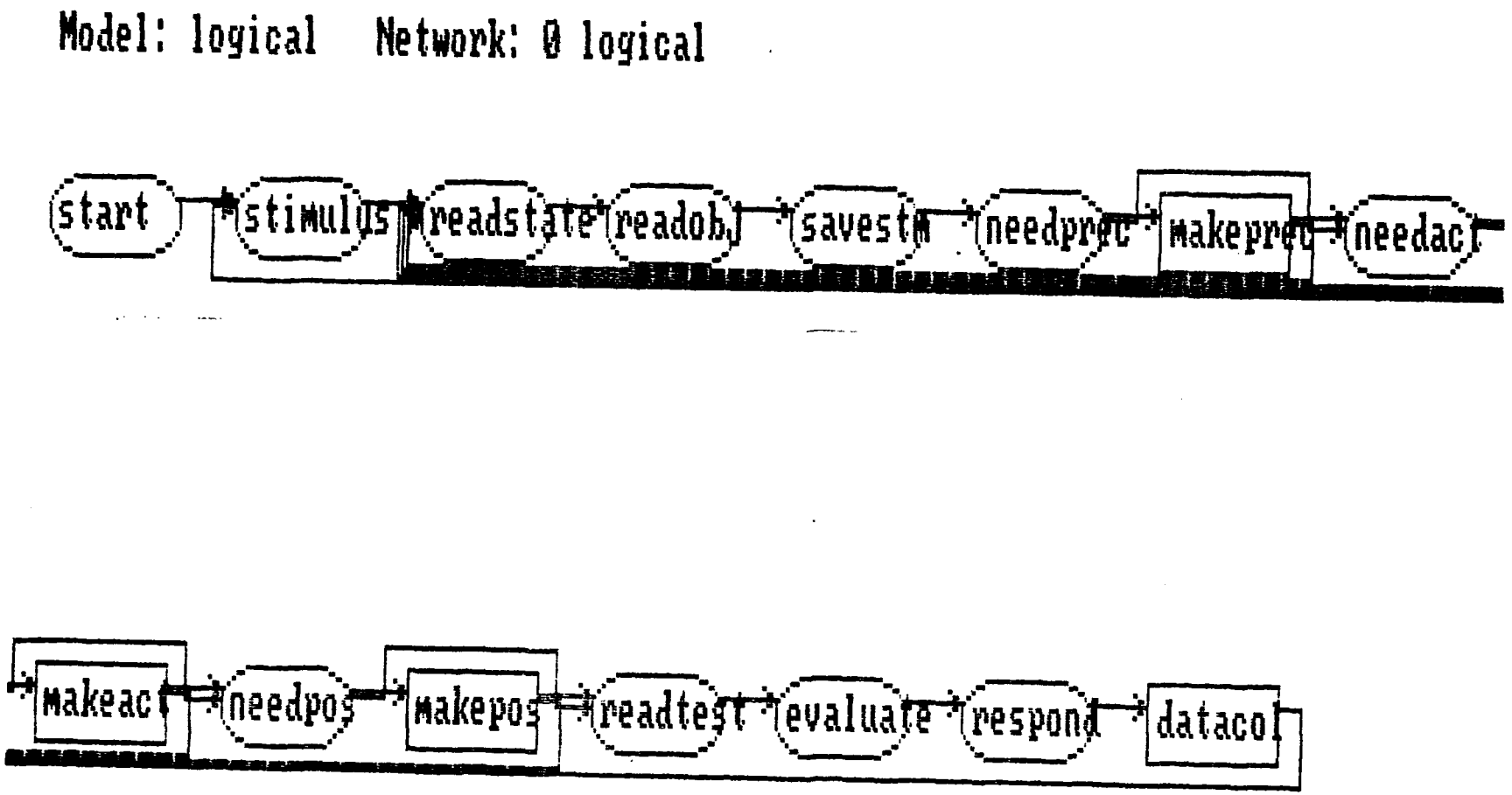


FIGURE 6

A MicroSAINT network diagram printout for the Logical model

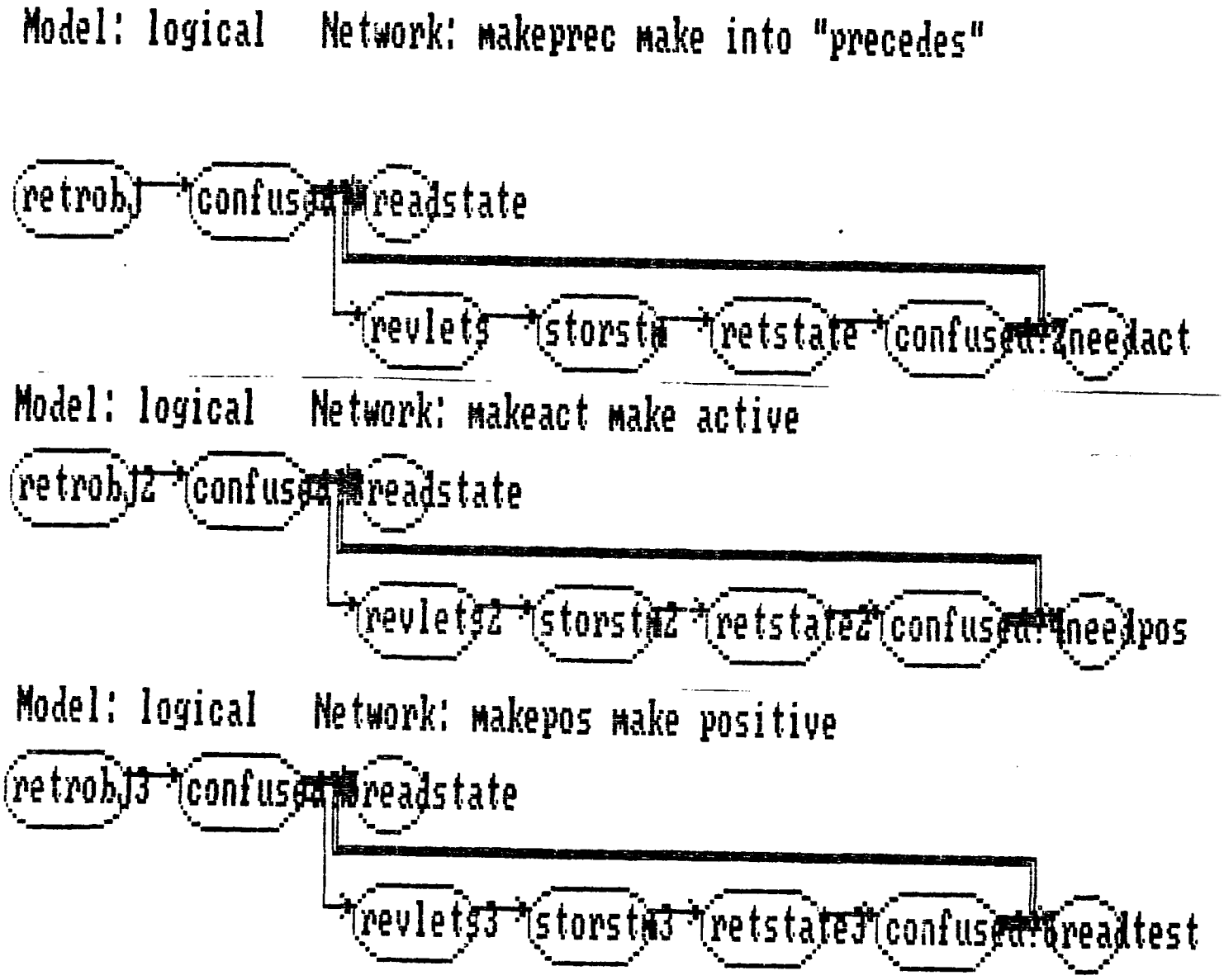


FIGURE 7

A MicroSAIN subnetworks diagrams for the Logical model

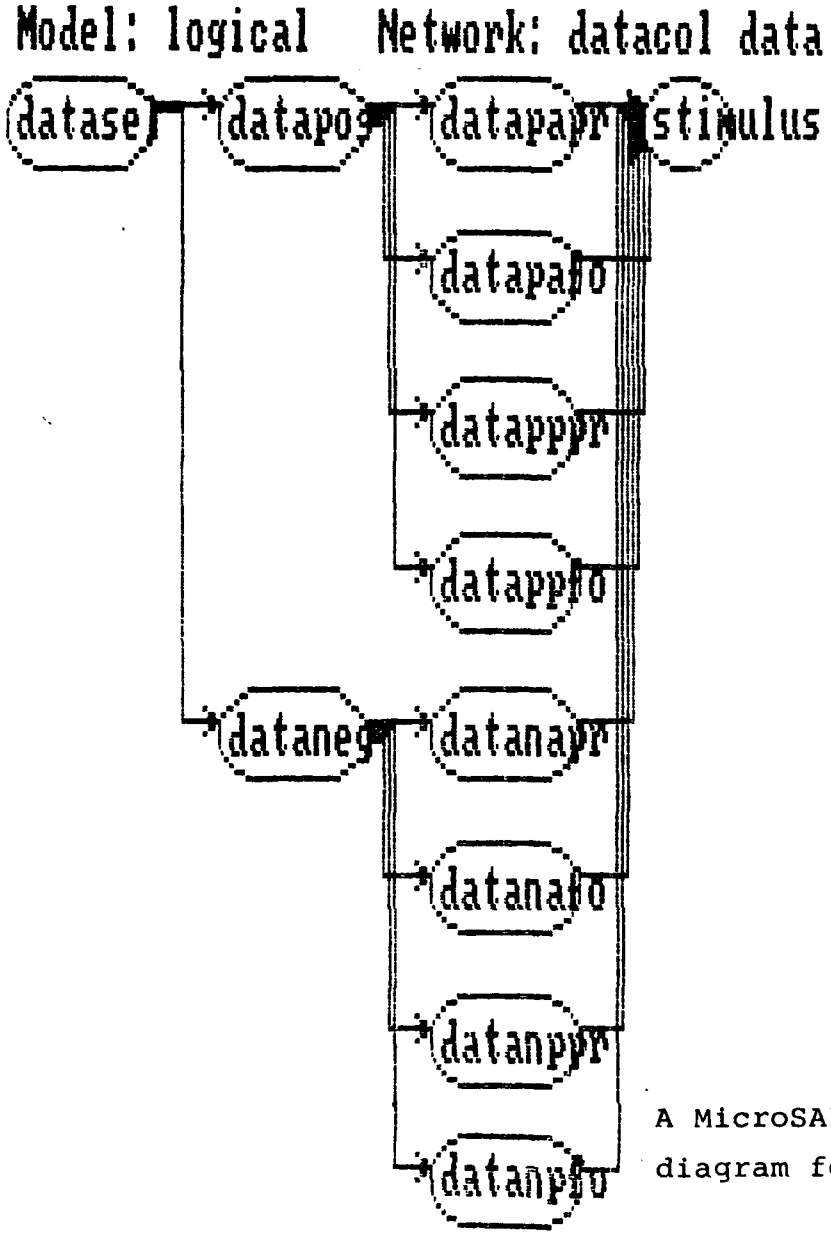


FIGURE 8

A MicroSAIN data collection subnetwork diagram for the Logical model

IV. THE LOGICAL REASONING TEST MODEL

Introduction

The Logical Reasoning Test requires the subject to interpret a statement about the sequence of the letters A and B and determine if this sequence is the same as the presented sequence. The nature of the statements requires that the subject "decode" the statement to determine the sequence described. For the purposes of this model, we have assumed that the subject will use a decoding strategy which converts the statement into a positive statement, using active voice and the verb root precede. These conversions are represented by switching the order of the letters in the statement (i.e., A does not follow B becomes B follows A; A is preceded by B becomes B precedes A; B follows A becomes A precedes B).

It is possible to make errors at each of the conversions. These errors may result in the subject becoming confused and having to reread the statement or they may result in the subject making an error in the conversion. In the first case, the error will result in a time penalty for performance. In the second case an error will occur. The likelihood of this second type of error is determined in the model by using a cumulative probability of error. Each time a conversion is made, the likelihood of error is compounded. The determination of whether or not an error occurred is at the end of the model and is based on this cumulative probability.

The basic sequence of events for the model is as follows. The stimulus is presented and the subject reads the statement and the order of the letters in the statement. He then makes conversions in the following order as needed: 1) precede verb form, 2) active voice, 3) positive statement. Errors may occur at each of these conversions and the subject will return to reading the statement if they do. Once the statement is converted the subject then reads the test sequence and compares it to the converted sequence. The subject then gives his response. This completes the model run.

Organization for the Documentation

This document is a mix of model printouts from Micro SAINT and word-processed explanations. It begins with the network diagram and the listing of the tasks. It then proceeds to the individual task descriptions. This is generally a page for page mix of Micro SAINT printouts and explanations. At the end of this are the Micro SAINT printouts of the Simulation Scenario, Snapshots, Functions, and Continuous Variable Changes as appropriate. These are referenced in the explanations.

Task Descriptions

In this section an explanation of each of the tasks in the model is provided. In these explanations, the variables, equations, and logic used in the task are described as well as the relationship of the task to other tasks in the model. In addition, this model contains subnetworks for making the conversions and for performing the data collection. These subnetworks are discussed in the order in which they occur and the tasks within them are identified with the additional heading "NETWORK NUMBER NETWORK NAME".

All times are in milliseconds. The function rand() returns a random number from a uniform distribution.

TASK NETWORK

Network Number: 0
(1) Name: logical (2) Type: Network
(3) Upper Network:
(4) Release Condition: 1;
(5) First sub-job: start start simulation
(6) Sub-jobs (each can be task or network):

Number:	Name:	Type:
start	start simulation	Task
stimulus	present stimulus	Task
readstate	read statement	Task
readobj	read objects	Task
savestm	save into STM	Task
makeprec	make into "precedes"	Network
needprec	preceed conversion?	Task
needact	active conversion?	Task
makeact	make active	Network
makepos	make positive	Network
needpos	negative conversion?	Task
evaluate	evaluate	Task
readtest	read test statement	Task
respond	make response	Task
datacol	data collection	Network

TASK NUMBER TASK NAME

start start simulation

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are currently set to 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

Upon the completion of this task the present stimulus task is initiated.

Task Number: start
(1) Name: start simulation *(2) Type: Task*
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect: nrespt=0;totrespt=0;totright=0;
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) stimulus	presen	(12)	1;
(13)		(14)	
(15)		(16)	
(17)		(18)	
(19)		(20)	
(21)		(22)	
(23)		(24)	

The nrespt (number of responses), accumulative response time, and total number of right responses are initialized in this task.

TASK NUMBER TASK NAME

stimulus present stimulus

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are currently set to 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

The ending effect of this task determines the type of statement which is presented as a stimulus. This is done by using three variables to represent the three characteristics of the statement which can be manipulated. The variables, their possible values, and the meaning of the values are presented below:

Variable	Value	Meaning
stateact	1	active voice
stateact	0	passive voice
statepos	1	positive statement
statepos	0	negative statement
statepre	1	precede verb form
statepre	0	follows verb form

Any time the value of one of these variables is 0 a conversion will be performed to change the statement to the desired form. The values for these variables are determined randomly using the following expressions:

```
if rand() < .5 then stateact = 0 else stateact = 1;
if rand() < .5 then statepos = 0 else statepos = 1;
if rand() < .5 then statepre = 0 else statepre = 1;
```

FOLLOWING TASKS

Upon the completion of this task the read statement task is initiated.

```
Task Number:  stimulus
(1)  Name:  present stimulus                    (2)  Type:  Task
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
if rand() < .5 then stateact = 0 else stateact = 1;
if rand() < .5 then statepos = 0 else statepos = 1;
if rand() < .5 then statepre = 0 else statepre = 1;
stimtime=clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  readstate     read s  (12)  1;
(13) .                      (14)
```

TASK NUMBER TASK NAME

readstate read statement

MEAN TIME AND STANDARD DEVIATION

The time to read the statement will be affected by its form. Positive statements using an active voice are read more quickly than the other statements. Given this, a test is performed to determine if the statement is in the positive-active voice format. If it is a time of 230 is used as the mean time. If not a time of 460 is used. The standard deviation is 80 for both cases. The expression used to perform this test is:

```
if stateact == 1 & statepos == 1 then 230 else 460;
```

BEGINNING EFFECTS

None.

ENDING EFFECTS

Two things occur in the ending effect. First the variable "statecor" which is used for tracking the cumulative probability that the statement was read or recalled correctly is set equal to the value of "epread". "epread" contains the probability that reading occurred without error and has been set to .92 in the simulation scenario.

The second thing which occurs is to set the variable timest equal to the current clock value. This records the time at which the statement is read.

FOLLOWING TASKS

The read objects task is initiated next.

```
Task Number:  readstate
(1)  Name:  read statement                          (2)  Type:  Task
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  if stateact==1 & statepos==1 then 230 else 460;
(7)  Standard deviation:  80;
(8)  Task's beginning effect:
(9)  Task's ending effect:  statecor = epread;
timest=clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  readobj       read o  (12)  1;
(13)                        (14)
```

TASK NUMBER TASK NAME

readobj read objects

MEAN TIME AND STANDARD DEVIATION

The mean time to read the objects (A...B or B...A) is 460 and the standard deviation is 100.

BEGINNING EFFECTS

None.

ENDING EFFECTS

Two things occur in the ending effect. First the variable "objcor" which is used for tracking the cumulative probability that the objects was read or recalled correctly is set equal to the value of "epobj". "epobj" contains the probability that reading occurred without error and has been set to .94 in the simulation scenario.

The second thing which occurs is to set the variable timeobj equal to the current clock value. This records the time at which the objects are read.

FOLLOWING TASKS

The save into STM task is initiated next.

```
Task Number:  readobj
(1)  Name:  read objects                                  (2)  Type:  Task
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  460;
(7)  Standard deviation:  100;
(8)  Task's beginning effect:
(9)  Task's ending effect:  timeobj=clock;objcor = epobj;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  savestm       save i  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

TASK NUMBER	TASK NAME
savestm	save into STM

MEAN TIME AND STANDARD DEVIATION

A time of 100 is used for saving the information into short term memory. The standard deviation used is 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"statecor" and "objcor", the variables for recording the cumulative probabilities for correctly reading and recalling the statement and the objects are compounded by multiplying them by the value of "epstm". "epstm" contains the probability that there are no errors in short term memory function and has been set to .97 in the simulation scenario.

FOLLOWING TASKS

The precede conversion task is initiated next.

```
Task Number:  savestm
(1)  Name:  save into STM                          (2)  Type:  Task
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  100;
(7)  Standard deviation:  0;
(8)  Task's beginning effect:
(9)  Task's ending effect:  statecor = statecor * epstm;
objcor = objcor * epstm;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  needprec      precee  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

```
Network Number:  makeprec
(1)  Name:  make into "precedes"                    (2)  Type:  Network
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  First sub-job:  retrobj  retrieve obj STM
(6)  Sub-jobs (each can be task or network):
Number:         Name:                   Type:
retrobj         retrieve obj STM        Task
confused?       excessive confusion?    Task
revlets         reverse letters         Task
storstm         store into STM          Task
retstate        retrieve statement      Task
confused?2      excessive confusion?    Task
```

NETWORK NUMBER NETWORK NAME

makeprec make into "precedes"

TASK NUMBER TASK NAME

retobj retrieve obj STM

MEAN TIME AND STANDARD DEVIATION

The variable "stmrtim" contains the mean time to retrieve the objects from short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmrtim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epstm". "epstm" contains the probability that there are no errors in the short term memory functions and has been set to .97 in the simulation scenario.

FOLLOWING TASKS

The excessive confusion? task is initiated next.

```
Task Number:  retrobj
(1)  Name:  retrieve obj STM                    (2)  Type:  Task
(3)  Upper Network:  makeprec  make into "precedes"
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmrtim;
(7)  Standard deviation:  stmrtim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epstm;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  confused?     excess  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

makeprec make into "precedes"

TASK NUMBER TASK NAME

confused? excessive confusion?

MEAN TIME AND STANDARD DEVIATION

The mean time for the subject to determine whether or not he has become confused is 92. The standard deviation is 30. There are three possible paths which may be taken out of this task. They reflect the likelihood that the subject became confused about either the statement or the objects. While the subject may become confused about either one of these independently, he will return to the read statement task if either type of confusion occurs. If no confusion occurs he will proceed to the reverse letters task. The likelihood that the subject will not confuse the statement or the objects has been defined as a negative exponential function for this model. The variables "decvals" and "decvalo" reflect this likelihood for the statement and the objects respectively. The values for decvals and decvalo are computed as continuous variable changes updated every 50 msec using the following expressions:

```
decvals = (2.71828^(-1*(clock-timest)/decstate));
decvalo = (2.71828^(-1*(clock-timeobj)/decobj));
```

The expressions "clock-timest" and "clock-timeobj" reflect the time since the statement and object were stored. "decstate" and "decobj" determine the shape of the decay curves and have been set to 4000 and 8000 respectively in the simulation scenario.

Which of the three paths that is taken is determined probabilistically. The Micro SAINT software will take the values of "1-decvals", "1-decvalo", and 1, scale them, and determine a path randomly based on the relative proportions of each of the values.

Task Number: confused?
(1) Name: excessive confusion? (2) Type: Task
(3) Upper Network: makeprec make into "precedes"
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 92;
(7) Standard deviation: 30;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Probabilistic

	Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11)	readstate	read s	(12)	1-decvals;
(13)	readstate	read s	(14)	1-decvalo;
(15)	revlets	revers	(16)	1;

NETWORK NUMBER NETWORK NAME

makeprec make into "precedes"

TASK NUMBER TASK NAME

revlets reverse letters

MEAN TIME AND STANDARD DEVIATION

The variable "timepre" contains the mean time to make the precede verb form transformation. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing timepre by 3.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "eppre". "eppre" contains the probability that there are no errors in the precede verb form transformation and has been set to .9 in the simulation scenario.

FOLLOWING TASKS

The store into STM task is initiated next.

```
Task Number:  revlets
(1)  Name:  reverse letters                    (2)  Type:  Task
(3)  Upper Network:  makeprec  make into "precedes"
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  timepre;
(7)  Standard deviation:  timepre/3;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * eppre;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  storstm       store   (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

makeprec make into "precedes"

TASK NUMBER TASK NAME

storstm store into STM

MEAN TIME AND STANDARD DEVIATION

The variable "stmstim" contains the mean time to store the objects into short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmstim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epstm". "epstm" contains the probability that there are no errors in the short term memory functions and has been set to .97 in the simulation scenario.

The time the objects have been in memory is also reinitialized by setting the variable timeobj equal to the current clock value.

FOLLOWING TASKS

The retrieve statement task is initiated next.

```
Task Number:  storstm
(1)  Name:  store into STM                          (2)  Type:  Task
(3)  Upper Network:  makeprec  make into "precedes"
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmstim;
(7)  Standard deviation:  stmstim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epstm;
timeobj = clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  retstate      retrie  (12)  1;
(13)                        (14)
(15)                        (16)
```

NETWORK NUMBER NETWORK NAME

makeprec make into "precedes"

TASK NUMBER TASK NAME

retstate retrieve statement

MEAN TIME AND STANDARD DEVIATION

The variable "stmrtim" contains the mean time to retrieve the statement into short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmstim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The second excessive confusion task is initiated next.

Task Number: retstate
(1) Name: retrieve statement (2) Type: Task
(3) Upper Network: makeprec make into "precedes"
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: stmrtim;
(7) Standard deviation: stmrtim/2;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) confused?2	*excess*	*(12)*	*1;*
(13)		*(14)*	
(15)		*(16)*	
(17)		*(18)*	
(19)		*(20)*	
(21)		*(22)*	
(23)		*(24)*	

NETWORK NUMBER NETWORK NAME

makeprec make into "precedes"

TASK NUMBER TASK NAME

confused?2 excessive confusion?

MEAN TIME AND STANDARD DEVIATION

The mean time for the subject to determine whether or not he has become confused is 92. The standard deviation is 30.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The function of the following tasks for this task are identical to that for the first confused task except that the active conversion task is substituted for the reverse letters task.

```
Task Number:  confused?2
(1)  Name:  excessive confusion?                    (2)  Type:  Task
(3)  Upper Network:  makeprec  make into "precedes"
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  92;
(7)  Standard deviation:  30;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Probabilistic
     Following Task/Network:        Probability Of Taking
      Number:       Name:            This Path:
(11)  readstate     read s  (12)  1-decvals;
(13)  readstate     read s  (14)  1-decvalo;
(15)  needact       active  (16)  1;
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

TASK NUMBER TASK NAME

needprec precede conversion?

MEAN TIME AND STANDARD DEVIATION

The amount of time needed to determine whether or not it is necessary to convert the statement into the precede verb form will be affected by whether the statement is positive and uses active voice. If either one of these is not the case then the decision time will be longer.

The base time for making the decision is 47. If the statement is negative or the statement uses passive voice an additional time of 47 is added for each of the cases. The total mean time is then reflected in the value of the variable "time". The expressions used to compute this mean time are as follows:

```
time = 47;
if stateact < 1 then time = time + 47;
if statepos < 1 then time = time + 47;
time;
```

The standard deviation for the task is 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

There are two possible paths which may be taken out of this task. If the value of the variable "statepre" is 0, then the subnetwork make into "precedes" will be entered. If the value of "statepre" is 1, then the statement is already in this form and the task active conversion is initiated.

Task Number: needprec
(1) Name: preceed conversion? *(2) Type: Task*
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: time = 47;
if stateact < 1 then time = time + 47;
if statepos < 1 then time = time + 47;
time;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Tactical

Following Task/Network: Number:	Name:		Tactical Expression:
(11) makeprec	make 1	(12)	statepre == 0;
(13) needact	active	(14)	statepre == 1;
(15)		(16)	

TASK NUMBER TASK NAME

needact active conversion?

MEAN TIME AND STANDARD DEVIATION

The amount of time needed to determine whether or not it is necessary to convert the statement into active voice form will be affected by whether the statement is positive and uses the precede verb form. If either one of these is not the case then the decision time will be longer.

The base time for making the decision is 47. If the statement is negative or the statement uses the follow verb form an additional time of 47 is added for each of the cases. The total mean time is then reflected in the value of the variable "time". The expressions used to compute this mean time are as follows:

```
time = 47;
if statepre < 1 then time = time + 47;
if statepos < 1 then time = time + 47;
time;
```

The standard deviation for the task is 0.

BEGINNING EFFECTS
None.

ENDING EFFECTS
None.

FOLLOWING TASKS
There are two possible paths which may be taken out of this task. If the value of the variable "stateact" is 0, then the subnetwork make into "active" will be entered. If the value of "stateact" is 1, then the statement is already in this form and the task negative conversion is initiated.

Task Number: needact
(1) Name: active conversion? *(2) Type: Task*
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: time = 47;
if statepre < 1 then time = time + 47;
if statepos < 1 then time = time + 47;
time;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Tactical

Following Task/Network: Number:	Name:		Tactical Expression:
(11) makeact	*make a*	*(12)*	*stateact == 0;*
(13) needpos	*negati*	*(14)*	*stateact == 1;*
(15)		*(16)*	

```
Network Number:  makeact
(1)  Name:  make active                          (2)  Type:  Network
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  First sub-job:  retrobj2  retrieve ob STM
(6)  Sub-jobs (each can be task or network):
Number:        Name:                 Type:
retrobj2       retrieve ob STM       Task
confused?3     confusion?            Task
revlets2       reverse letters       Task
storstm2       store into STM        Task
retstate2      retrieve statement    Task
confused?4     confused?             Task
```

NETWORK NUMBER NETWORK NAME

makeact make active

TASK NUMBER TASK NAME

retobj2 retrieve obj STM

MEAN TIME AND STANDARD DEVIATION

The variable "stmrtim" contains the mean time to retrieve the objects from short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmrtim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epstm". "epstm" contains the probability that there are no errors in the short term memory functions and has been set to .97 in the simulation scenario.

FOLLOWING TASKS

The confusion? task is initiated next.

```
Task Number:  retrobj2
(1)  Name:  retrieve ob SIM                          (2)  Type:  Task
(3)  Upper Network:  makeact  make active
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmrtim;
(7)  Standard deviation:  stmrtim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epstm;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  confused?3    confus  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

makeact make active

TASK NUMBER TASK NAME

confused?3 confusion?

MEAN TIME AND STANDARD DEVIATION

The mean time for the subject to determine whether or not he has become confused is 92. The standard deviation is 30.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

There are three possible paths which may be taken out of this task. They reflect the likelihood that the subject became confused about either the statement or the objects. While the subject may become confused about either one of these independently, he will return to the read statement task if either type of confusion occurs. If no confusion occurs he will proceed to the reverse letters task.

The likelihood that the subject will not confuse the statement or the objects has been defined as a negative exponential function for this model. The variables "decvals" and "decvalo" reflect this likelihood for the statement and the objects respectively. The values for decvals and decvalo are computed as continuous variable changes updated every 50 msec using the following expressions:

```
decvals = (2.71828^(-1*(clock-timest)/decstate));
decvalo = (2.71828^(-1*(clock-timeobj)/decobj));
```

The expressions "clock-timest" and "clock-timeobj" reflect the time since the statement and object were stored. "decstate" and "decobj" determine the shape of the decay curves and have been set to 4000 and 8000 respectively in the simulation scenario.

Which of the three paths that is taken is determined probabilistically. The Micro SAINT software will take the values of "1-decvals", "1-decvalo", and 1, scale them, and determine a path randomly based on the relative proportions of each of the values.

```
Task Number:  confused?3
(1)  Name:  confusion?                                  (2)  Type:  Task
(3)  Upper Network:  makeact  make active
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  92;
(7)  Standard deviation:  30;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Probabilistic
     Following Task/Network:        Probability Of Taking
      Number:        Name:           This Path:
(11)  readstate      read s  (12)  1-decvals;
(13)  readstate      read s  (14)  1-decvalo;
(15)  revlets2       revers  (16)  1;
(17)                         (18)
(19)                         (20)
(21)                         (22)
(23)                         (24)
```

NETWORK NUMBER NETWORK NAME

makeact make active

TASK NUMBER TASK NAME

revlets2 reverse letters

MEAN TIME AND STANDARD DEVIATION

The variable "timeact" contains the mean time to make active voice form transformation. It is currently set to 400 in the simulation scenario. The standard deviation is computed by dividing timeact by 3.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epact". "epact" contains the probability that there are no errors in the active voice form transformation and has been set to .95 in the simulation scenario.

FOLLOWING TASKS

The store into STM task is initiated next.

```
Task Number:  revlets2
(1)  Name:  reverse letters                    (2)  Type:  Task
(3)  Upper Network:  makeact  make active
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  timeact;
(7)  Standard deviation:  timeact/3;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epact;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  storstm2      store   (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

makeact make active

TASK NUMBER TASK NAME

storstm2 store into STM

MEAN TIME AND STANDARD DEVIATION

The variable "stmstim" contains the mean time to store the objects into short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmstim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epstm". "epstm" contains the probability that there are no errors in the short term memory functions and has been set to .97 in the simulation scenario.

The time the objects have been in memory is also reinitialized by setting the variable timeobj equal to the current clock value.

FOLLOWING TASKS

The retrieve statement task is initiated next.

```
Task Number:  storstm2
(1)  Name:  store into STM                          (2)  Type:  Task
(3)  Upper Network:  makeact  make active
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmstim;
(7)  Standard deviation:  stmstim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epstm;
timeobj = clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11) retstate2     retrie  (12)  1;
(13)                       (14)
(15)                       (16)
```

NETWORK NUMBER NETWORK NAME

makeact make active

TASK NUMBER TASK NAME

retstate2 retrieve statement

MEAN TIME AND STANDARD DEVIATION

The variable "stmrtim" contains the mean time to retrieve the statement into short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmstim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The second excessive confusion task is initiated next.

Task Number: retstate2
(1) Name: retrieve statement (2) Type: Task
(3) Upper Network: makeact make active
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: stmrtim;
(7) Standard deviation: stmrtim/2;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:	Probability Of Taking This Path:
(11) confused?4	confus (12)	1;
(13)	(14)	
(15)	(16)	
(17)	(18)	
(19)	(20)	
(21)	(22)	
(23)	(24)	

NETWORK NUMBER NETWORK NAME

makeact make active

TASK NUMBER TASK NAME

confused?4 excessive confusion?

MEAN TIME AND STANDARD DEVIATION

The mean time for the subject to determine whether or not he has become confused is 92. The standard deviation is 30.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The function of the following tasks for this task are identical to that for the first confused task except that the negative conversion task is substituted for the reverse letters task.

Task Number: confused?4
(1) Name: confused? *(2) Type: Task*
(3) Upper Network: makeact make active
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 92;
(7) Standard deviation: 30;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Probabilistic

	Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11)	readstate	read s	(12)	1-decvals;
(13)	readstate	read s	(14)	1-decvalo;
(15)	needpos	negati	(16)	1;
(17)			(18)	
(19)			(20)	
(21)			(22)	
(23)			(24)	

TASK NUMBER TASK NAME

needpos negative conversion?

MEAN TIME AND STANDARD DEVIATION

The amount of time needed to determine whether or not it is necessary to convert the statement into positive form will be affected by whether the statement uses active voice and the precede verb form. If either one of these is not the case then the decision time will be longer.

The base time for making the decision is 47. If the statement is passive or the statement uses the follow verb form an additional time of 47 is added for each of the cases. The total mean time is then reflected in the value of the variable "time". The expressions used to compute this mean time are as follows:

```
time = 47;
if stateact < 1 then time = time + 47;
if statepre < 1 then time = time + 47;
time;
```

The standard deviation for the task is 0.

FOLLOWING TASKS

There are two possible paths which may be taken out of this task. If the value of the variable "statepos" is 0, then the subnetwork make into "positive" will be entered. If the value of "statepos" is 1, then the statement is already in this form and the task read test is initiated.

```
Task Number: needpos
(1) Name: negative conversion?                    (2) Type: Task
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: time = 47;
if stateact < 1 then time = time + 47;
if statepre < 1 then time = time + 47;
time;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Tactical
     Following Task/Network:      Tactical Expression:
      Number:      Name:
(11) makepos      make p  (12)  statepos == 0;
(13) readtest     read t  (14)  statepos == 1;
```

```
Network Number:  makepos
(1)  Name:  make positive                          (2)  Type:  Network
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  First sub-job:  retrobj3  retrieve ob STM
(6)  Sub-jobs (each can be task or network):
Number:        Name:                  Type:
retrobj3       retrieve ob STM        Task
confused?5     confusion?             Task
revlets3       reverse letters        Task
storstm3       store into STM         Task
retstate3      retrieve statement     Task
confused?6     confused?              Task
```

NETWORK NUMBER NETWORK NAME

makepos make positive

TASK NUMBER TASK NAME

retobj3 retrieve obj STM

MEAN TIME AND STANDARD DEVIATION

The variable "stmrtim" contains the mean time to retrieve the objects from short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmrtim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epstm". "epstm" contains the probability that there are no errors in the short term memory functions and has been set to .97 in the simulation scenario.

FOLLOWING TASKS

The confusion? task is initiated next.

```
Task Number:  retrobj3
(1)  Name:  retrieve ob STM                          (2)  Type:  Task
(3)  Upper Network:  makepos  make positive
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmrtim;
(7)  Standard deviation:  stmrtim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epstm;
(10) Decision Type:  Single choice
     Following Task/Network:        Probability Of Taking
      Number:       Name:            This Path:
(11)  confused?5    confus  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

makepos make positive

TASK NUMBER TASK NAME

confused?5 confusion?

MEAN TIME AND STANDARD DEVIATION

The mean time for the subject to determine whether or not he has become confused is 92. The standard deviation is 30.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

There are three possible paths which may be taken out of this task. They reflect the likelihood that the subject became confused about either the statement or the objects. While the subject may become confused about either one of these independently, he will return to the read statement task if either type of confusion occurs. If no confusion occurs he will proceed to the reverse letters task.

The likelihood that the subject will not confuse the statement or the objects has been defined as a negative exponential function for this model. The variables "decvals" and "decvalo" reflect this likelihood for the statement and the objects respectively. The values for decvals and decvalo are computed as continuous variable changes updated every 50 msec using the following expressions:

```
decvals = (2.71828^(-1*(clock-timest)/decstate));
decvalo = (2.71828^(-1*(clock-timeobj)/decobj));
```

The expressions "clock-timest" and "clock-timeobj" reflect the time since the statement and object were stored. "decstate" and "decobj" determine the shape of the decay curves and have been set to 4000 and 8000 respectively in the simulation scenario.

Which of the three paths that is taken is determined probabilistically. The Micro SAINT software will take the values of "1-decvals", "1-decvalo", and 1, scale them, and determine a path randomly based on the relative proportions of each of the values.

```
Task Number:  confused?5
(1)  Name:  confusion?                                  (2)  Type:  Task
(3)  Upper Network:  makepos  make positive
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  92;
(7)  Standard deviation:  30;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Probabilistic
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  readstate     read s  (12)  1-decvals;
(13)  readstate     read s  (14)  1-decvalo;
(15)  revlets3      revers  (16)  1;
(17)                        (18)
(19)                        (20)
(21)                        (22)
```

NETWORK NUMBER NETWORK NAME

makepos make positive

TASK NUMBER TASK NAME

revlets3 reverse letters

MEAN TIME AND STANDARD DEVIATION

The variable "timepos" contains the mean time to make positive form transformation. It is currently set to 500 in the simulation scenario. The standard deviation is computed by dividing timepos by 3.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "eppos". "eppos" contains the probability that there are no errors in the positive form transformation and has been set to .9 in the simulation scenario.

FOLLOWING TASKS

The store into STM task is initiated next.

```
Task Number:  revlets3
(1)  Name:  reverse letters                          (2)  Type:  Task
(3)  Upper Network:  makepos  make positive
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  timepos;
(7)  Standard deviation:  timepos/3;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * eppos;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  storstm3      store   (12)  1;
(13)                        (14)
(15)                        (16)
```

NETWORK NUMBER NETWORK NAME

makepos make positive

TASK NUMBER TASK NAME

storstm3 store into STM

MEAN TIME AND STANDARD DEVIATION

The variable "stmstim" contains the mean time to store the objects into short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmstim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

"objcor", the variable for recording the cumulative probability for correctly reading and recalling the objects is compounded by multiplying it by the value of "epstm". "epstm" contains the probability that there are no errors in the short term memory functions and has been set to .97 in the simulation scenario.

The time the objects have been in memory is also reinitialized by setting the variable timeobj equal to the current clock value.

FOLLOWING TASKS

The retrieve statement task is initiated next.

```
Task Number:  storstm3
(1)  Name:  store into STM                          (2)  Type:  Task
(3)  Upper Network:  makepos  make positive
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmstim;
(7)  Standard deviation:  stmstim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:  objcor = objcor * epstm;
timeobj = clock;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  retstate3     retrie  (12)  1;
```

NETWORK NUMBER	NETWORK NAME
makepos	make positive

TASK NUMBER	TASK NAME
retstate3	retrieve statement

MEAN TIME AND STANDARD DEVIATION

The variable "stmrtim" contains the mean time to retrieve the statement into short term memory. It is currently set to 100 in the simulation scenario. The standard deviation is computed by dividing stmstim by 2.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The second excessive confusion task is initiated next.

```
Task Number:  retstate3
(1)  Name:  retrieve statement                      (2)  Type:  Task
(3)  Upper Network:  makepos  make positive
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  stmrtim;
(7)  Standard deviation:  stmrtim/2;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  confused?6    confus  (12)  1;
```

NETWORK NUMBER NETWORK NAME

makepos make positive

TASK NUMBER TASK NAME

confused?6 excessive confusion?

MEAN TIME AND STANDARD DEVIATION

The mean time for the subject to determine whether or not he has become confused is 92. The standard deviation is 30.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The function of the following tasks for this task are identical to that for the first confused task except that the read test task is substituted for the reverse letters task.

Task Number: confused?6
(1) Name: confused? *(2) Type: Task*
(3) Upper Network: makepos make positive
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 92;
(7) Standard deviation: 30;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Probabilistic

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) readstate	read s	(12)	1-decvals;
(13) readstate	read s	(14)	1-decvalo;
(15) readtest	read t	(16)	1;
(17)		(18)	
(19)		(20)	
(21)		(22)	
(23)		(24)	

TASK NUMBER TASK NAME

readtest read test statement

MEAN TIME AND STANDARD DEVIATION

The mean and standard deviation for reading the test letters (i.e., AB or BA) are 230 and 30 respectively.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The evaluate task is initiated next.

```
Task Number: readtest
(1) Name: read test statement                    (2) Type: Task
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 230;
(7) Standard deviation: 30;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  evaluate      evalua  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

TASK NUMBER TASK NAME

evaluate evaluate

MEAN TIME AND STANDARD DEVIATION

The mean and standard deviation for comparing the letter sequence converted from the statement to the test letters are 80 and 16 respectively.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The make response task is initiated next.

Task Number: evaluate
(1) Name: evaluate *(2) Type: Task*
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 80;
(7) Standard deviation: 16;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) respond	make r	*(12)*	1;
(13)		*(14)*	
(15)		*(16)*	
(17)		*(18)*	
(19)		*(20)*	
(21)		*(22)*	
(23)		*(24)*	

TASK NUMBER TASK NAME

respond make response

MEAN TIME AND STANDARD DEVIATION

The mean and standard deviation for making the response are 150 and 20 respectively.

BEGINNING EFFECTS

None.

ENDING EFFECTS

The cumulative probability that an error was not made contained in the variable "objcor" is compared to a random number to determine if an error occurred. If no error occurred then the variable "response" is set to 1. If an error occurred then "response" is set to 0.

FOLLOWING TASKS

The data collection network is entered next.

The number of responses (nrespt), average response time (avrespt), and percent right are computed after each trial.

```
Task Number:  respond
(1)  Name:  make response                          (2)  Type:  Task
(3)  Upper Network:  0  logical
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  150;
(7)  Standard deviation:  20;
(8)  Task's beginning effect:
(9)  Task's ending effect:
if rand() < objcor then response = 1 else response = 0;
resptime=clock-stimtime;
nrespt=nrespt+1;totrespt=totrespt+resptime;avrespt=totrespt/nrespt;
totright=totright+response;totwrong=nrespt-totright;
perright=100*totright/nrespt;
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  datacol       data c  (12)  1;
(13)                        (14)
(15)                        (16)
```

Network Number: datacol
(1) Name: data collection (2) Type: Network
(3) Upper Network: 0 logical
(4) Release Condition: 1;
(5) First sub-job: datasel data collection
(6) Sub-jobs (each can be task or network):

Number:	Name:	Type:
datapos	positive statements	Task
dataneg	negative statements	Task
datapapr	pos/active/precedes	Task
datapafo	pos/active/follows	Task
datapppr	positive/passive/pre	Task
datasel	data collection	Task
datappfo	pos/passive/follows	Task
datanapr	neg/active/precedes	Task
datanafo	neg/active/follows	Task
datanppr	neg/passive/precedes	Task
datanpfo	neg/passive/follows	Task

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datasel data collection

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The function of this data collection subnetwork is to allow for separate snapshot files to be made for each of the possible combinations of positive/negative - passive/active - precede/follow statements. This is done by successively testing the values of the variables statepos, stateact, and statepre. This results in ending in one of eight possible tasks for each of the combinations. A snapshot is taken when one of these tasks ends.

This task is the first step in the process and determines of the statement is positive or negative by evaluating statepos. If statepos is 1 then the positive statements task is initiated, otherwise the negative statements task is initiated.

```
Task Number: datasel
(1)  Name: data collection                          (2)  Type: Task
(3)  Upper Network: datacol  data collection
(4)  Release Condition: 1;
(5)  Time Distribution Type: Normal
(6)  Mean Time: 0;
(7)  Standard deviation: 0;
(8)  Task's beginning effect:
(9)  Task's ending effect:
(10) Decision Type: Tactical
     Following Task/Network:       Tactical Expression:
      Number:      Name:
(11) datapos      positi  (12)  statepos == 1;
(13) dataneg      negati  (14)  statepos < 1;
(15)                      (16)
```

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datapos positive statements

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The possible combinations of stateact and statepre are tested to determine the complete nature of the statement. The combinations, the statement meaning, and the following task are outlined below.

Combination stateact	statepre	Statement	Following Task
1	1	positive/active/precede	datapapr
1	0	positive/active/follow	datapafo
0	1	positive/passive/precede	datapppr
0	0	positive/passive/follow	datappfo

Task Number: datapos
(1) Name: positive statements (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Tactical

Following Task/Network: Number:	Name:		Tactical Expression:
(11) datapapr	pos/ac	(12)	stateact == 1 & statepre == 1;
(13) datapafo	pos/ac	(14)	stateact == 1 & statepre < 1;
(15) datapppr	positi	(16)	stateact < 1 & statepre == 1;
(17) datappfo	pos/pa	(18)	stateact < 1 & statepre <1;
(19)		(20)	
(21)		(22)	
(23)		(24)	

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

dataneg negative statements

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

The possible combinations of stateact and statepre are tested to determine the complete nature of the statement. The combinations, the statement meaning, and the following task are outlined below.

Combination stateact	statepre	Statement	Following Task
1	1	negative/active/precede	datanapr
1	0	negative/active/follow	datanafo
0	1	negative/passive/precede	datanppr
0	0	negative/passive/follow	datanpfo

Task Number: dataneg
(1) Name: negative statements (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Tactical

Following Task/Network: Number:	Name:		Tactical Expression:
(11) datanapr	neg/ac	(12)	stateact == 1 & statepre == 1;
(13) datanafo	neg/ac	(14)	stateact == 1 & statepre < 1;
(15) datanppr	neg/pa	(16)	stateact < 1 & statepre == 1;
(17) datanpfo	neg/pa	(18)	stateact < 1 & statepre <1;

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datapapr pos/active/precedes

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "papr" and contains the variables "response" and "objcor".

Task Number: datapapr
(1) Name: pos/active/precedes (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) stimulus	presen	(12)	1;
(13)		(14)	
(15)		(16)	
(17)		(18)	
(19)		(20)	
(21)		(22)	
(23)		(24)	

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datapafo pos/active/follows

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "pafo" and contains the variables "response" and "objcor".

Task Number: datapafo
(1) Name: pos/active/follows (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11) stimulus	presen	(12)	1;
(13)		(14)	
(15)		(16)	
(17)		(18)	
(19)		(20)	
(21)		(22)	

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datapppr pos/passive/pre

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "pppr" and contains the variables "response" and "objcor".

Task Number: datapppr
(1) Name: positive/passive/pre (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

Following Task/Network: Number:	Name:	Probability Of Taking This Path:
(11) stimulus	presen (12)	1;
(13)	(14)	
(15)	(16)	
(17)	(18)	
(19)	(20)	
(21)	(22)	
(23)	(24)	

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datappfo pos/passive/follows

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "ppfo" and contains the variables "response" and "objcor".

Task Number: datappfo
(1) Name: pos/passive/follows (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

	Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11)	stimulus	presen	(12)	1;
(13)			(14)	
(15)			(16)	
(17)			(18)	
(19)			(20)	
(21)			(22)	
(23)			(24)	

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datanapr neg/active/precedes

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "napr" and contains the variables "response" and "objcor".

Task Number: datanapr
(1) Name: neg/active/precedes (2) Type: Task
(3) Upper Network: datacol data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice

	Following Task/Network: Number:	Name:		Probability Of Taking This Path:
(11)	stimulus	presen	(12)	1;
(13)			(14)	
(15)			(16)	
(17)			(18)	
(19)			(20)	
(21)			(22)	
(23)			(24)	

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datanafo neg/active/follows

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "nafo" and contains the variables "response" and "objcor".

```
Task Number:  datanafo
(1)  Name:  neg/active/follows                    (2)  Type:  Task
(3)  Upper Network:  datacol  data collection
(4)  Release Condition:  1;
(5)  Time Distribution Type:  Normal
(6)  Mean Time:  0;
(7)  Standard deviation:  0;
(8)  Task s beginning effect:
(9)  Task's ending effect:
(10) Decision Type:  Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stimulus      presen  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datanppr neg/passive/pre

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "nppr" and contains the variables "response" and "objcor".

```
Task Number: datanppr
(1) Name: neg/passive/precedes                    (2) Type: Task
(3) Upper Network: datacol  data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stimulus      presen  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

NETWORK NUMBER NETWORK NAME

datacol data collection

TASK NUMBER TASK NAME

datanpfo neg/passive/follows

MEAN TIME AND STANDARD DEVIATION

The mean time and standard deviation for this task are 0.

BEGINNING EFFECTS

None.

ENDING EFFECTS

None.

FOLLOWING TASKS

None. The snapshot file initiated at the end of this task is called "npfo" and contains the variables "response" and "objcor".

```
Task Number: datanpfo
(1) Name: neg/passive/follows                    (2) Type: Task
(3) Upper Network: datacol  data collection
(4) Release Condition: 1;
(5) Time Distribution Type: Normal
(6) Mean Time: 0;
(7) Standard deviation: 0;
(8) Task's beginning effect:
(9) Task's ending effect:
(10) Decision Type: Single choice
     Following Task/Network:      Probability Of Taking
      Number:       Name:          This Path:
(11)  stimulus      presen  (12)  1;
(13)                        (14)
(15)                        (16)
(17)                        (18)
(19)                        (20)
(21)                        (22)
(23)                        (24)
```

SNAPSHOTS OF EXECUTION

```
( 1)  Trigger:           End trigger
( 2)  Task/Network:      datapapr       pos/active/precedes
( 6)  Snapshot file:     papr
Variables to store:
( 7)  clock                         ( 8)  response
( 9)  objcor                        (10)
(11)                                (12)
(13)                                (14)
(15)                                (16)


( 1)  Trigger:           End trigger
( 2)  Task/Network:      datapafo       pos/active/follows
( 6)  Snapshot file:     pafo
Variables to store:
( 7)  clock                         ( 8)  response
( 9)  objcor                        (10)
(11)                                (12)
(13)                                (14)
(15)                                (16)


( 1)  Trigger:           End trigger
( 2)  Task/Network:      datapppr       positive/passive/pre
( 6)  Snapshot file:     pppr
Variables to store:
( 7)  clock                         ( 8)  response
( 9)  objcor                        (10)
(11)                                (12)
(13)                                (14)
(15)                                (16)


( 1)  Trigger:           End trigger
( 2)  Task/Network:      datappfo       pos/passive/follows
( 6)  Snapshot file:     ppfo
Variables to store:
( 7)  clock                         ( 8)  response
( 9)  objcor                        (10)
(11)                                (12)
(13)                                (14)
(15)                                (16)
```

```
( 1)  Trigger:            End trigger
( 2)  Task/Network:       datanppr      neg/passive/precedes
( 6)  Snapshot file:      nppr
Variables to store:
( 7)  clock                          ( 8)  response
( 9)  objcor                         (10)
(11)                                 (12)
(13)                                 (14)
(15)                                 (16)


( 1)  Trigger:            End trigger
( 2)  Task/Network:       datanpfo      neg/passive/follows
( 6)  Snapshot file:      npfo
Variables to store:
( 7)  clock                          ( 8)  response
( 9)  objcor                         (10)
(11)                                 (12)
(13)                                 (14)
(15)                                 (16)


( 1)  Trigger:            End trigger
( 2)  Task/Network:       datacol       data collection
( 6)  Snapshot file:      dataall
Variables to store:
( 7)  clock                          ( 8)  response
( 9)  objcor                         (10)  stateact
(11)  statepos                       (12)  statepre
(13)                                 (14)
(15)                                 (16)


( 1)  Trigger:            End trigger
( 2)  Task/Network:       datanafo      neg/active/follows
( 6)  Snapshot file:      nafo
Variables to store:
( 7)  clock                          ( 8)  response
( 9)  objcor                         (10)
(11)                                 (12)
(13)                                 (14)
(15)                                 (16)


( 1)  Trigger:            End trigger
( 2)  Task/Network:       datanapr      neg/active/precedes
( 6)  Snapshot file:      napr
Variables to store:
```

```
( 1)  Trigger:          End trigger
( 2)  Task/Network:     respond        make response



( 6)  Snapshot file:    logsum
Variables to store:
( 7)  nrespt                          ( 8)  resptime
( 9)  avrespt                         (10)  totwrong
(11)  perright                        (12)
(13)                                  (14)
(15)                                  (16)
```

SIMULATION SCENARIO

```
( 1)  Event Time:          0.00
( 2)  Expression:          epact = .95;
timepos = 500;
timeact = 400;
timepre = 100;
eppos = .9;
eppre = .9;
epread = .92;
epobj = .94;
epstm = .97;
stmrtim = 100;
stmstim = 100;
decstate = 4000;
decobj = 8000;
```

CONTINUOUS VARIABLE CHANGES

```
( 1)  Event Time:          10.00
( 2)  Expression:
decvals = (2.71828 ^ (-1*(clock -timest)/decstate));
decvalo = (2.71828 ^ (-1*(clock-timeobj)/decobj));
( 3)  Update Interval:     50.000
( 4)  Stop Time:           Continue forever
```

Sample Output From Logical Model

The following table represents a sample output from the logical reasoning model. Test time was set at 3 minutes. We included results from the first and last five trials. The total number of trials is 27. The average response time and the percentage right are 6576.3 MS and 66.67%, respectively.

TABLE 6

Model: logical Network: 2 logical

"clock" "nrespt" "resptime" "avrespt" "totwrong" "perright" "Trigger:" "Jobs
2652.668701 1 2652.668701 2652.668701 1 0.000000 "End" "respond"
4232.603027 2 1579.934326 2116.301514 1 50.000000 "End" "respond"
5736.180176 3 1503.577148 1912.060059 1 66.666664 "End" "respond"
9549.882812 4 3813.702637 2387.470703 2 50.000000 "End" "respond"
15446.832031 5 5896.949219 3089.366455 2 60.000000 "End" "respond"

148683.468750 23 9144.578125 6464.498535 7 69.565216 "End" "respond"
160489.343750 24 11805.875000 6687.056152 8 66.666664 "End" "respond"
164113.093750 25 3623.750000 6564.523926 9 64.000000 "End" "respond"
167419.343750 26 3306.250000 6439.205566 9 65.384613 "End" "respond"
177561.984375 27 10142.640625 6576.369629 9 66.666664 "End" "respond"

V. DISCUSSION

As mentioned previously, the models represented by the simulation programs of Sections II through IV are preliminary. for example, the default values of the model parameters that can be found in the program documentation are working values. Some of these were derived from the collection of rule of thumb estimates of mental process parameters collected and published by Card, Moran, and Newell [2]. Others currently have the status of frank guesses. hence, all are regarded as very tentative and we do not claim they are accurate.

We are currently evaluating all of the models for theoretical and empirical adequacies. Our initial work in this direction will involve fitting the models to empirical data, carrying out sensitivity analyses of model parameters and performing tests of the models' assumptions.

We expect to substantially revise all of the models. For example, the exponentially shaped visual excitation function used in the choice reaction time model was selected somewhat arbitrarily. The choice of the exponential was not without precedent; however, we are currently evaluating several alternatives.

The decision to incorporate noise at the level of the perceptual system in the reaction time model was also somewhat arbitrary (but, again, not without precedent). An alternative that seems more consistent with introspection, at least, would be to add noise to the memory system in the form of trial to trial variation in the response criterion. It seems likely that doing this would not directly affect the model's predictions. However, adopting this approach would allow the model to account for response slowing attributable to stressors that affect memory systems but not the visual system.

Motor system activation and priming were not considered in this version of the reaction time model, nor in any of the models discussed here. Such phenomena probably will have to be incorporated if the models are to deal adequately with sequential effects such as perseveration and, perhaps, some types of impulsivity.

The reader may have noted that logical reasoning model assumes that subjects identify and transform the grammatical structures of the stimulus propositions serially. This process could be carried out in parallel fashion. We plan to examine the possibility that it does, empirically, and amend the model accordingly. We also plan to revise the forgetting function employed in the logical reasoning model to more accurately

eflect interference among traces and changes in trace strengths that occur during reading and recall. As in the case of the reaction time model, the motor system representation of logical reasoning model may benefit from elaboration.

Finally, it seems unlikely that the manikin models described here exhaust the range of strategies subjects employ in this task. Particularly worrisome are the possibilities that subjects select mixed strategies in this task--or switch between strategies when stressed. As was the case in the reaction time and logical reasoning models, the forgetting function and motor system representation employed in the manikin model warrant further development.

VI. REFERENCES

1. Bender, E.A., 1978, "An Introduction to Mathematical Modeling", John Wiley & Sons, Inc., 1978.

2. Card, S.K., Moran, H.F., and Newell, A., 1983, "The Psychology of Human - Computer Interaction", Lawrence Erlbaum Associates, Hillsdale, New Jersey,

3. Englund, C.E., R.L. Reeves, C.A. Shingledecker, D.R. Thorne, K.P. Wilson, and F.W. Hegge, 1985, "Unified Tri-Service Cognitive Performance Assessment Battery (UTC-PAB): Design and Specification of the Battery", JW6D3 MILPERF Report No. 87-10, Naval Medical Research and Development Command, Bethesda, MD.

4. Maki, D.E., M. Thompson, 1973, "Mathematical Models and Applications", Prentice Hall, Inc., Englewood Cliffs, New Jersey.

5. MicroSAINT User's Guide, 1987, Micro Analysis and Design, Boulder, CO.

6. Perez, W.A., P.J. Masline, E.G. Ramsey, K.E. Urbana, 1987, "Unified Tri-Service Cognitive Performance Assessment Battery: Review and Methodology (U)", Report No. AAMRL-TR-87-007, Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.